Complexity Study of Reasoning about Knowledge and Public Observations

Avijeet Ghosh[1]

[1]Indian Statistical Institute, Kolkata

Joint work with
Sourav Chakraborty, Sujata Ghosh, François Schwarzentruber

Table of Contents

Table of Contents

How to Model Real-Life Scenarios

How do we mathematically model certain scenarios?

How to Model Real-Life Scenarios

How do we mathematically model certain scenarios?

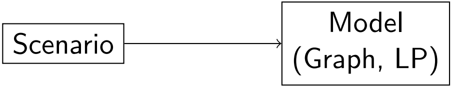
- Graphs, Flows, Linear Programs

$$\boxed{\text{Scenario}} \longrightarrow \boxed{\begin{array}{c}\text{Model}\\ \text{(Graph, LP)}\end{array}}$$

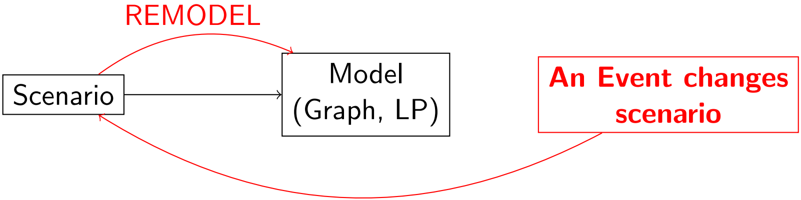
How to Model Real-Life Scenarios

How do we mathematically model certain scenarios?

- Graphs, Flows, Linear Programs



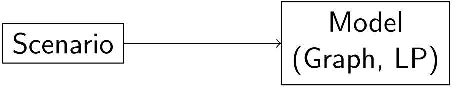
- **Problem**:



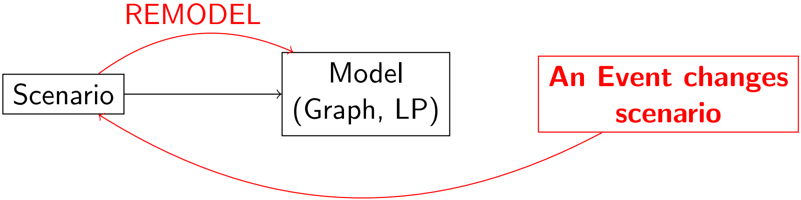
How to Model Real-Life Scenarios

How do we mathematically model certain scenarios?

- Graphs, Flows, Linear Programs



- **Problem**:

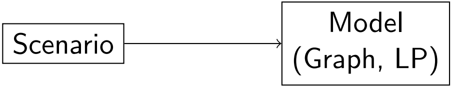


- Examples:

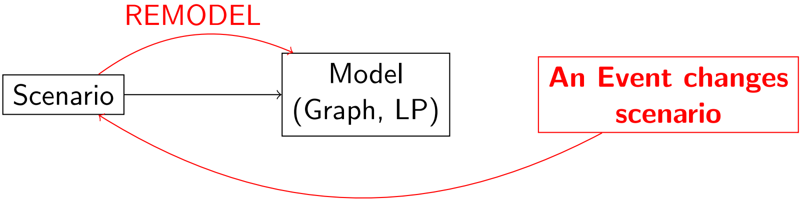
How to Model Real-Life Scenarios

How do we mathematically model certain scenarios?

- Graphs, Flows, Linear Programs



- **Problem**:

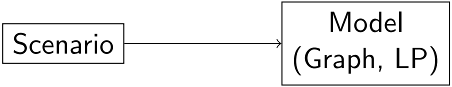


- Examples:
  - Supervisor Assignment Problem: Perfect Matching

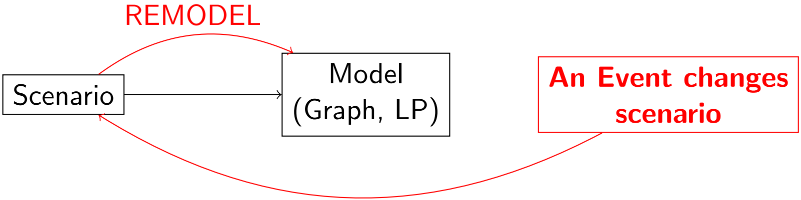
How to Model Real-Life Scenarios

How do we mathematically model certain scenarios?

- Graphs, Flows, Linear Programs



- **Problem**:



- Examples:
  - Supervisor Assignment Problem: Perfect Matching
  - Whether a certain town is reachable given a map: Reachability

Some More Examples

- Propositional Language (Valuation models):

$$\varphi_G \to \varphi_C$$

Some More Examples

- Propositional Language (Valuation models):

$$\varphi_G \to \varphi_C$$

- First-Order Language (Domain-Interpretation models):

$$\exists x \in \mathbb{N} : \ \forall y \in \mathbb{N} : \ (x \leq y)?$$

Some More Examples

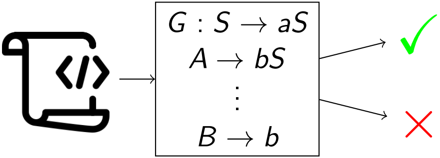
- Propositional Language (Valuation models):

$$\varphi_G \to \varphi_C$$

- First-Order Language (Domain-Interpretation models):

$$\exists x \in \mathbb{N} : \ \forall y \in \mathbb{N} : \ (x \leq y)?$$

- Context-Free Language: Is a certain program correct syntactically?



Some More Examples

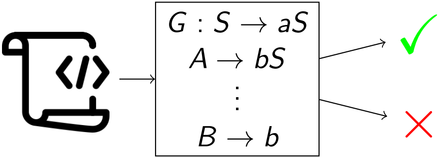
- Propositional Language (Valuation models):

$$\varphi_G \to \varphi_C$$

- First-Order Language (Domain-Interpretation models):

$$\exists x \in \mathbb{N} : \ \forall y \in \mathbb{N} : \ (x \leq y)?$$

- Context-Free Language: Is a certain program correct syntactically?



- Buchi Automata: Will an OS arrive at a deadlock EVENTUALLY?

Modelling Knowledge

- How about modelling **Knowledge**?

Modelling Knowledge

- How about modelling **Knowledge**?
- 3 people (A, B, C) picks one card each from a deck of 3 cards $(1, 2, 3)$.

Modelling Knowledge

- How about modelling **Knowledge**?
- 3 people (A, B, C) picks one card each from a deck of 3 cards $(1, 2, 3)$.
- How to consider **KNOWLEDGE OF AGENTS**?

Modelling Knowledge

- How about modelling **Knowledge**?
- 3 people (A, B, C) picks one card each from a deck of 3 cards $(1, 2, 3)$.
- How to consider **KNOWLEDGE OF AGENTS**?
  - **Considering possibilities**:

$$\boxed{\begin{array}{l} A > 3 \\ B > 2 \\ C > 1 \end{array}} \quad \boxed{\begin{array}{l} A > 1 \\ B > 2 \\ C > 3 \end{array}} \quad \boxed{\begin{array}{l} A > 2 \\ B > 3 \\ C > 1 \end{array}} \quad \cdots$$

Modelling Knowledge

- How about modelling **Knowledge**?
- 3 people (A, B, C) picks one card each from a deck of 3 cards $(1, 2, 3)$.
- How to consider **KNOWLEDGE OF AGENTS**?
  - **Considering possibilities**:

$$\boxed{\begin{array}{l} A > 3 \\ B > 2 \\ C > 1 \end{array}} \quad \boxed{\begin{array}{l} A > 1 \\ B > 2 \\ C > 3 \end{array}} \quad \boxed{\begin{array}{l} A > 2 \\ B > 3 \\ C > 1 \end{array}} \quad \cdots$$

  - **Indistinguishable Possibilities**:

For B: $\boxed{\begin{array}{l} A > 3 \\ B > 2 \\ C > 1 \end{array}} \overset{B}{\text{———}} \boxed{\begin{array}{l} A > 1 \\ B > 2 \\ C > 3 \end{array}}$, For A: $\boxed{\begin{array}{l} A > 3 \\ B > 2 \\ C > 1 \end{array}} \overset{A}{\text{———}} \boxed{\begin{array}{l} A > 3 \\ B > 1 \\ C > 2 \end{array}}$

Modelling Knowledge

- How about modelling **Knowledge**?
- 3 people (A, B, C) picks one card each from a deck of 3 cards $(1, 2, 3)$.
- How to consider **KNOWLEDGE OF AGENTS**?
  - **Considering possibilities**:

    $$\boxed{\begin{array}{l} A > 3 \\ B > 2 \\ C > 1 \end{array}} \quad \boxed{\begin{array}{l} A > 1 \\ B > 2 \\ C > 3 \end{array}} \quad \boxed{\begin{array}{l} A > 2 \\ B > 3 \\ C > 1 \end{array}} \quad \cdots$$

  - **Indistinguishable Possibilities**:

    For B: $\boxed{\begin{array}{l} A > 3 \\ B > 2 \\ C > 1 \end{array}} \overset{B}{\text{———}} \boxed{\begin{array}{l} A > 1 \\ B > 2 \\ C > 3 \end{array}}$, For A: $\boxed{\begin{array}{l} A > 3 \\ B > 2 \\ C > 1 \end{array}} \overset{A}{\text{———}} \boxed{\begin{array}{l} A > 3 \\ B > 1 \\ C > 2 \end{array}}$

  - **Event changes Knowledge**: A tells B it has 3, now for B: $\boxed{\begin{array}{l} A > 3 \\ B > 2 \\ C > 1 \end{array}}$

Checking Solution vs Finding a Solution

- Two kind of questions using Linear Programs:

Checking Solution vs Finding a Solution

- Two kind of questions using Linear Programs:

  Max $c^T x$
  s.t. $Ax \leq b$

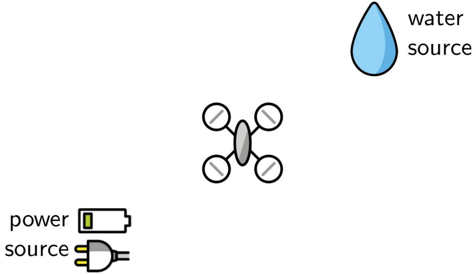
-
  $$x = \begin{bmatrix} 2 \\ 7 \\ \vdots \\ 3 \end{bmatrix}$$

  $\boxed{\text{Feasibility Check}} \longrightarrow$ Feasible?

Checking Solution vs Finding a Solution

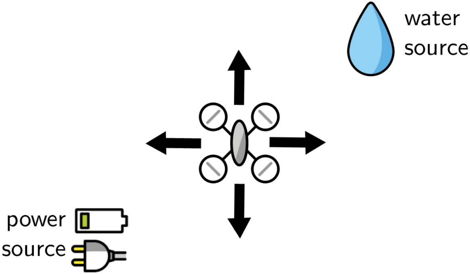
- Two kind of questions using Linear Programs:
  - Max $c^T x$
    s.t. $Ax \leq b$
    $x = \begin{bmatrix} 2 \\ 7 \\ \vdots \\ 3 \end{bmatrix}$ $\rightarrow$ $\boxed{\text{Feasibility Check}}$ $\rightarrow$ Feasible?
  - Max $c^T x$
    s.t. $Ax \leq b$ $\rightarrow$ $\boxed{\text{Feasible Solution Finding}}$ $\rightarrow$ $\exists$? a feasible solution

Motivation:A Farming Drone



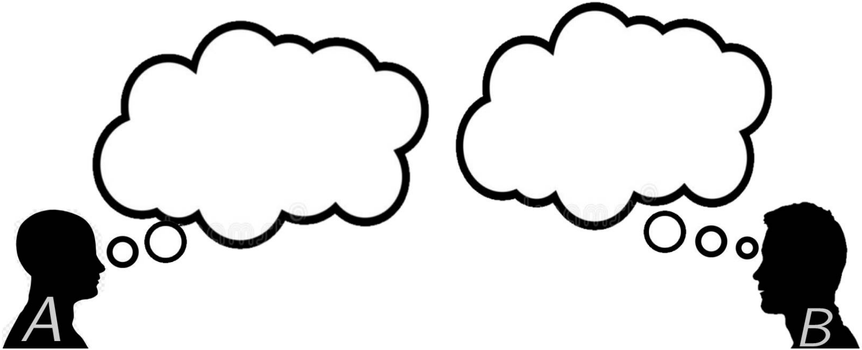
- A water source on top right corner
- A power source on bottom right corner

Motivation:A Farming Drone

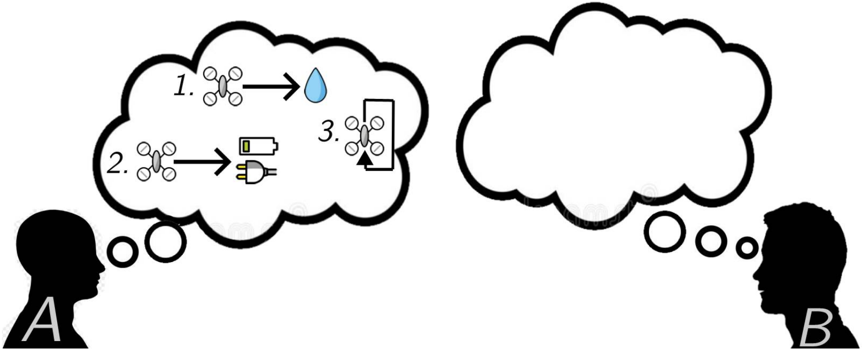


- Can move up, down, left or right
- Cannot move diagonally

Farming Drone: Agents and their expectations

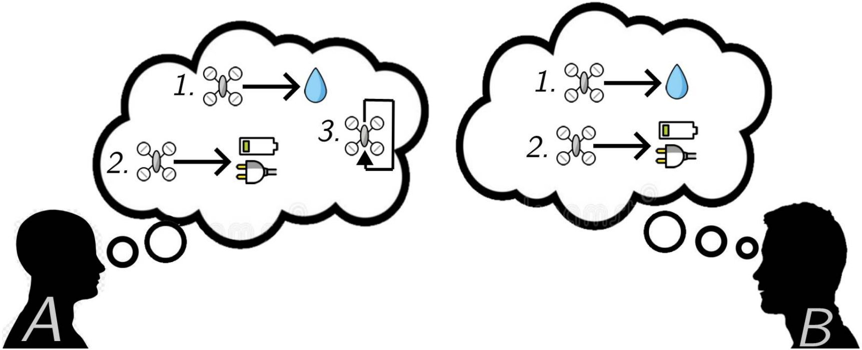


Farming Drone: Agents and their expectations



1. Go to water with ≤ 1 wrong move.
2. Go to power with ≤ 1 wrong move.
3. Go patrolling in clockwise direction.

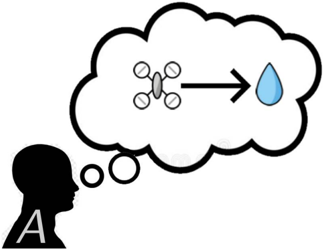
Farming Drone: Agents and their expectations



1. Go to water with ≤ 1 wrong move.
2. Go to power with ≤ 1 wrong move.
3. Go patrolling in clockwise direction.

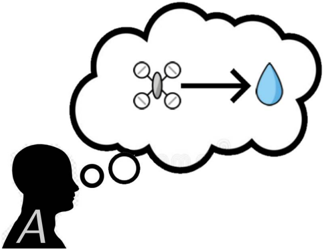
1. Go to water with ≤ 1 wrong move.
2. Go to power with ≤ 1 wrong move.

Farming Drone: Reasoning about this scenario



- What is the minimal number of  moves that A has to **observe** to know its goal?

Farming Drone: Reasoning about this scenario



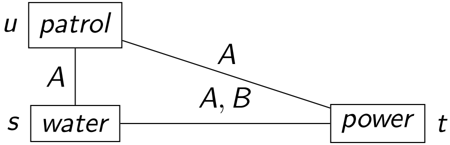
- What is the minimal number of  moves that A has to **observe** to know its goal?
- Does there exist a sequence of  moves such that by observing it, B would know its goal but A would not?

Modelling Knowledge: Epistemic Model

- Epistemic model (W, R, V).

[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

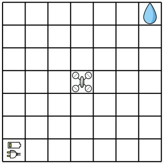
- Epistemic model (W, R, V).

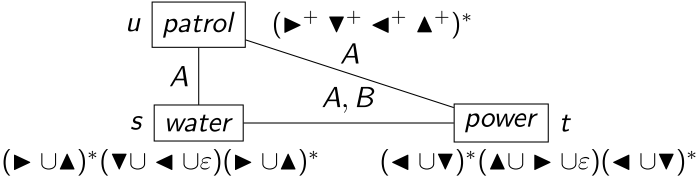


[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Modelling Observation: Epistemic Expectation Model[1]

- Epistemic model (W, R, V).
- Each world is assigned with a regular expression: a set of expected observations.

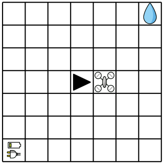


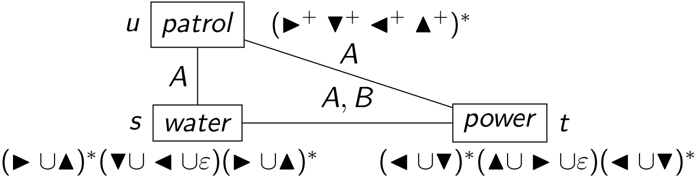


[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Modelling Observation: Epistemic Expectation Model[1]

- Epistemic model (W, R, V).
- Each world is assigned with a regular expression: a set of expected observations.
- Model can get truncated after a sequence of observation, say ▶.

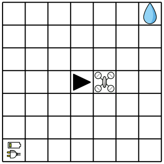


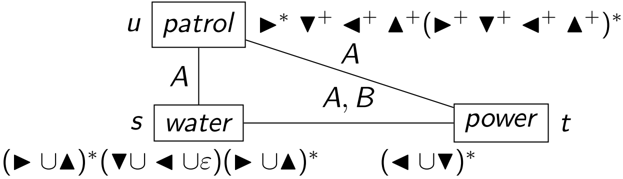


[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Modelling Observation: Epistemic Expectation Model[1]

- Epistemic model (W, R, V).
- Each world is assigned with a regular expression: a set of expected observations.
- Model can get truncated after a sequence of observation, say ▶.

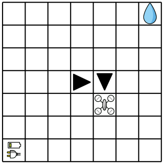


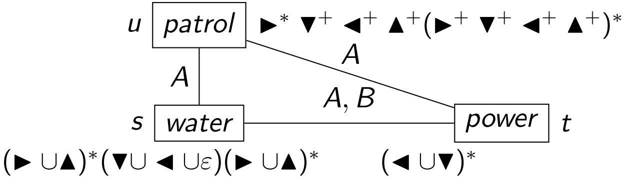


[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Modelling Observation: Epistemic Expectation Model[1]

- Epistemic model (W, R, V).
- Each world is assigned with a regular expression: a set of expected observations.
- Model can get truncated after a sequence of observation, say ▶ ▼.

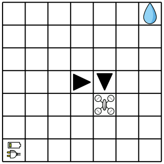


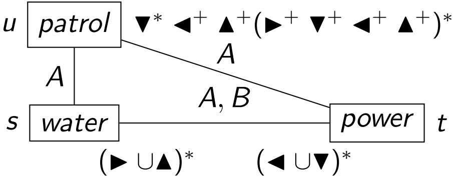


[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Modelling Observation: Epistemic Expectation Model[1]

- Epistemic model (W, R, V).
- Each world is assigned with a regular expression: a set of expected observations.
- Model can get truncated after a sequence of observation, say ▶ ▼.

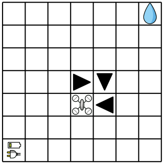


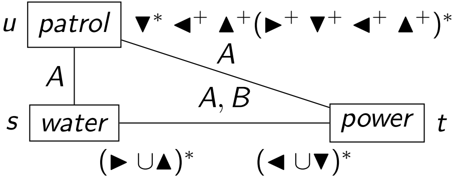


[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Modelling Observation: Epistemic Expectation Model[1]

- Epistemic model (W, R, V).
- Each world is assigned with a regular expression: a set of expected observations.
- Model can get truncated after a sequence of observation, say ▶ ▼ ◀.

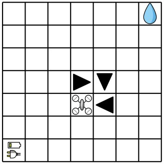




[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Modelling Observation: Epistemic Expectation Model[1]

- Epistemic model (W, R, V).
- Each world is assigned with a regular expression: a set of expected observations.
- Model can get truncated after a sequence of observation, say ▶ ▼ ◀.



u $\boxed{patrol}$ $◀^* ▲^+(▶^+ ▼^+ ◀^+ ▲^+)^*$

A

$\boxed{power}$ t

$(◀ \cup ▼)^*$

[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.

[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
    - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.

[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
  - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.

[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
  - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
  - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.

[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
  - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
  - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.
- For example, $\hat{K}_A$ water

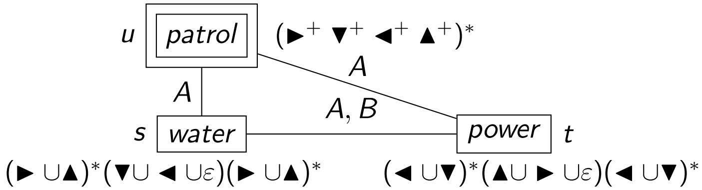
[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
    - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
    - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.
- For example, $\hat{K}_A\,\mathrm{water}$



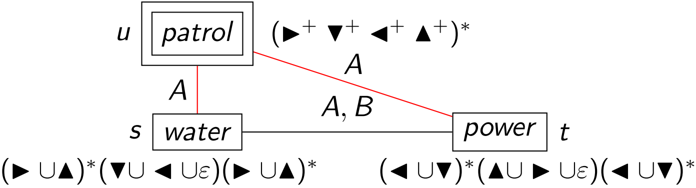
[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
  - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
  - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.
- For example, $\hat{K}_A$ water



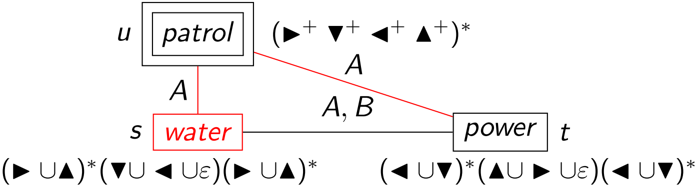
[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
  - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
  - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.
- For example, $\hat{K}_A$ water



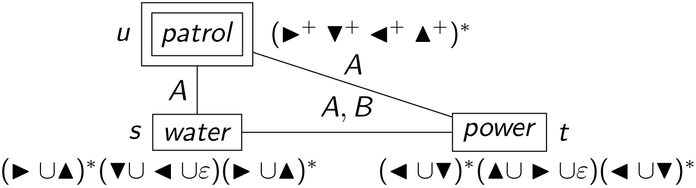
[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
  - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
  - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.
- For example, $\langle \blacktriangleright \blacktriangledown \blacktriangleleft \rangle \hat{K}_A \mathrm{water}$



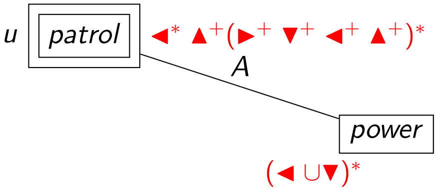
[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
    - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
    - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.
- For example, $\langle \blacktriangleright \blacktriangledown \blacktriangleleft \rangle \hat{K}_A\, \mathrm{water}$



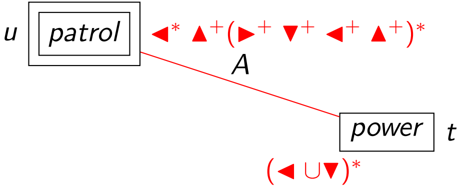
[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
  - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
  - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.
- For example, $\langle \blacktriangleright\ \blacktriangledown\ \blacktriangleleft \rangle \hat{K}_A\,\mathrm{water}$



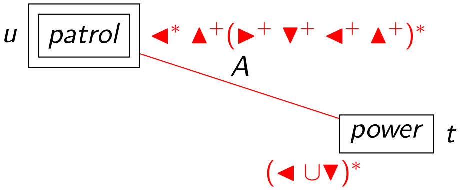
[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

Public Observation Logic: Syntax[1]

The language of POL:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \hat{K}_i\varphi \mid [\pi]\varphi \mid \langle\pi\rangle\varphi$$

- $K_i\varphi$: an agent i **knows** φ holds.
  - $\hat{K}_i\varphi$: an agent i considers φ **possibly** holds.
- $[\pi]\varphi$: after any sequence of **observation** matching π, φ holds.
  - $\langle\pi\rangle\varphi$: after some sequence of **observation** matching π, φ holds.
- For example, $\cancel{\langle \blacktriangleright \blacktriangledown \blacktriangleleft \rangle \hat{K}_A \text{ water}}$



[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014

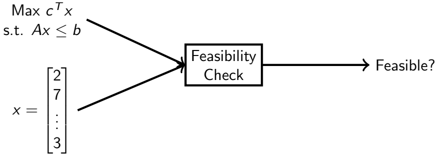
Table of Contents

Recall: Checking Solution vs Finding a Solution

- Recall the questions using Linear Programs:

Recall: Checking Solution vs Finding a Solution

- Recall the questions using Linear Programs:

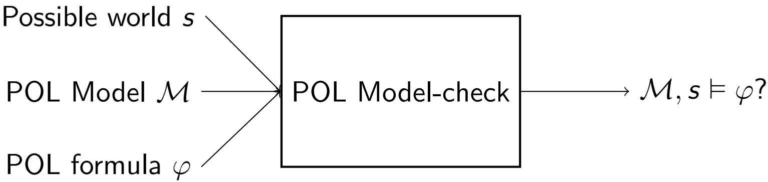
    - 

    : Check CNF Assignment

Recall: Checking Solution vs Finding a Solution

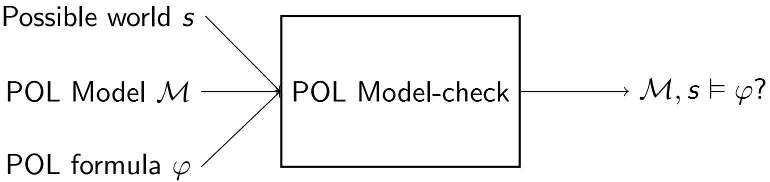
- Recall the questions using Linear Programs:
  - Max $c^T x$ s.t. $Ax \leq b$ and $x = \begin{bmatrix} 2 \\ 7 \\ \vdots \\ 3 \end{bmatrix}$ $\rightarrow$ $\boxed{\text{Feasibility Check}}$ $\rightarrow$ Feasible? : Check CNF Assignment
  - Max $c^T x$ s.t. $Ax \leq b$ $\rightarrow$ $\boxed{\text{Feasible Solution Finding}}$ $\rightarrow$ $\exists$? a feasible solution : CNF-SAT

The Model Checking Question



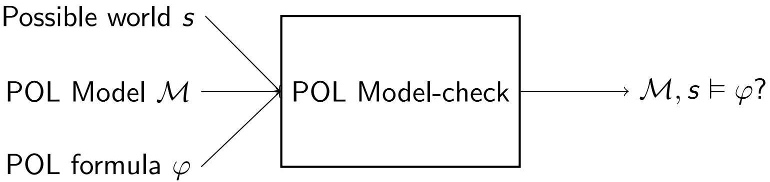
Is POL Model-checking **decidable**?

The Model Checking Question



Is POL Model-checking **decidable**?
Answer: Yes

The Model Checking Question

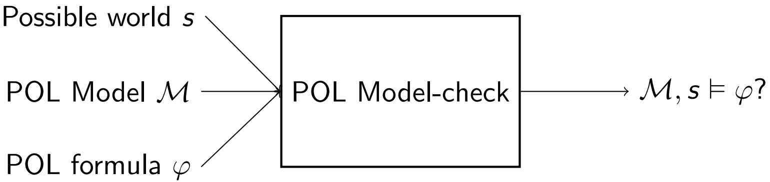


Is POL Model-checking **decidable**?
Answer: Yes

Theorem: POL Model Checking Complexity [IJCAI'22]

The model-checking problem of POL is PSPACE-Complete.

The Model Checking Question



Is POL Model-checking **decidable**?
Answer: Yes

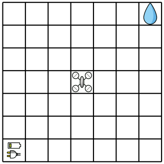
Theorem: POL Model Checking Complexity [IJCAI'22]

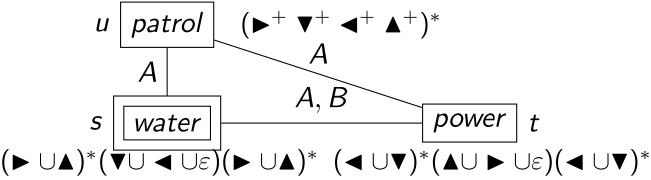
The model-checking problem of POL is PSPACE-Complete.

It's too hard, isn't it?
Enter: **Fragments**

POL Model-checking

Recall the example:

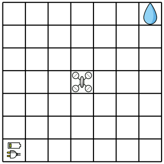


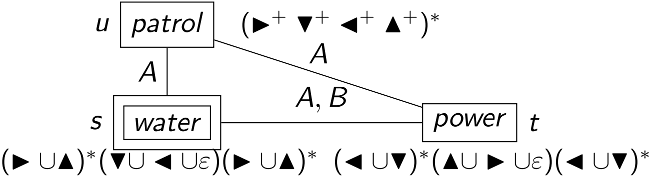


[1] T. Bolander, A Gentle Introduction to Epistemic Planning: The DEL Approach, M4M@ICLA 2017

POL Model-checking

Recall the example:



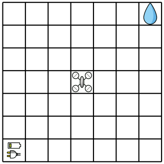


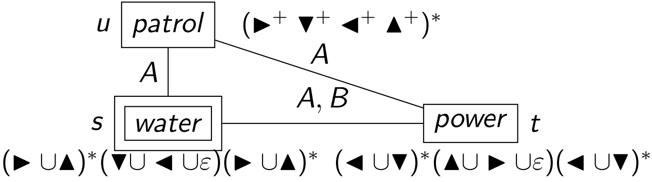
Question: Does there exist a sequence of -moves

[1] T. Bolander, A Gentle Introduction to Epistemic Planning: The DEL Approach, M4M@ICLA 2017

POL Model-checking

Recall the example:

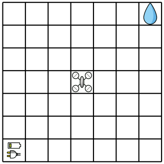


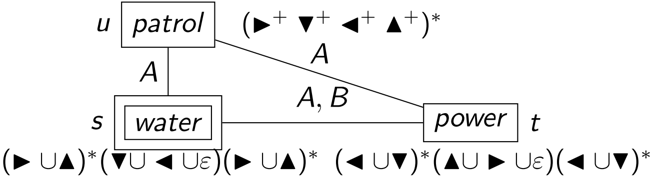


Question: Does there exist a sequence of -moves or a **PLAN**

[1] T. Bolander, A Gentle Introduction to Epistemic Planning: The DEL Approach, M4M@ICLA 2017

Recall the example:





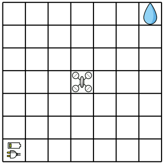
Question: Does there exist a sequence of -moves or a **PLAN**

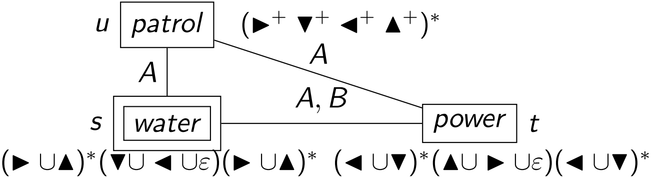
after which Knowledge of an agent changes? (Epistemic Planning[1])

[1] T. Bolander, A Gentle Introduction to Epistemic Planning: The DEL Approach, M4M@ICLA 2017

POL Model-checking

Recall the example:





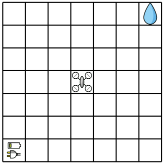
Question: Does there exist a sequence of -moves or a **PLAN**

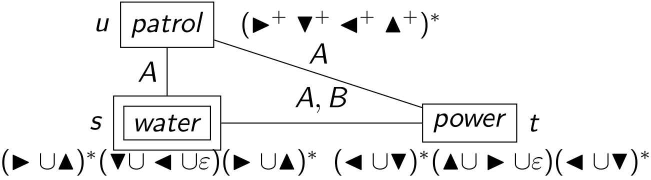
after which Knowledge of an agent changes? (Epistemic Planning[1])

Solution: $\mathcal{M}, s \vDash \langle(\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^{\star}\rangle K_A\varphi$

[1] T. Bolander, A Gentle Introduction to Epistemic Planning: The DEL Approach, M4M@ICLA 2017

Recall the example:



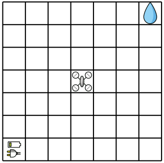


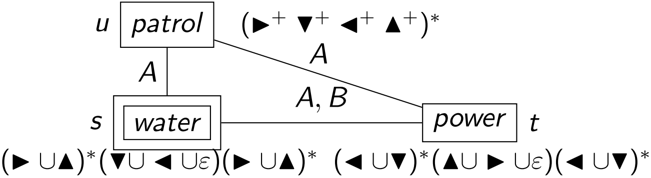
Question: Does there exist a sequence of -moves or a **PLAN** after which Knowledge of an agent changes? (Epistemic Planning[1])

Solution: $\mathcal{M}, s \vDash \langle(\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^{\star}\rangle K_A \varphi$ (**Model-Checking**)

[1] T. Bolander, A Gentle Introduction to Epistemic Planning: The DEL Approach, M4M@ICLA 2017

POL Model-checking

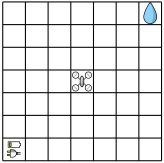


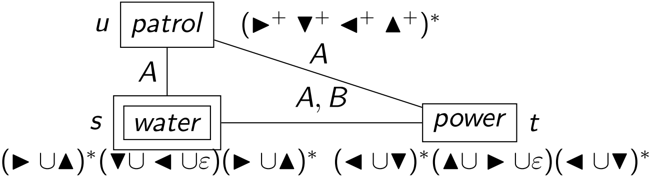


- **Verification of a plan,** Word **Fragment**: only **word** in π.

$$\mathcal{M}, s \models \langle \blacktriangleright\blacktriangleright\blacktriangleright \rangle K_A water$$

POL Model-checking



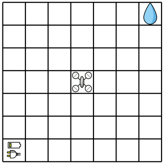


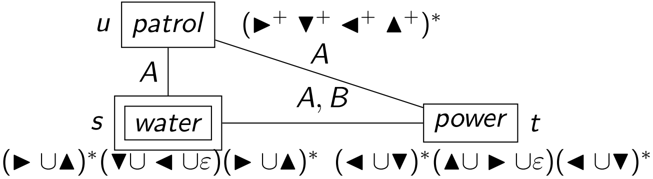
- **Verification of a plan,** Word **Fragment**: only **word** in π.

$$\mathcal{M}, s \models \langle \blacktriangleright\blacktriangleright\blacktriangleright \rangle K_A water$$

Can A know the goal is after observing **three** $\blacktriangleright$ **moves** (a plan)?

POL Model-checking

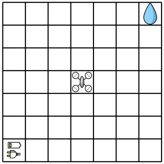


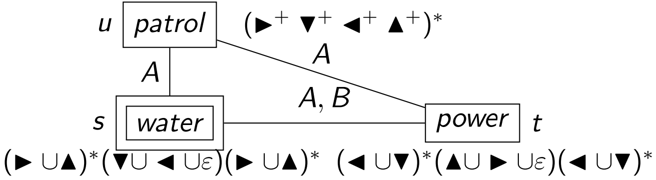


- Star-Free **Fragment**: no Kleene Star ($*$) in π.

$$\mathcal{M}, s \models [(\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^2] \neg K_A water \wedge \langle (\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^3 \rangle K_A water$$

POL Model-checking





- Star-Free **Fragment**: no Kleene Star $(*)$ in π.

$$\mathcal{M}, s \models [(\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^{2}] \neg K_{A} water \wedge \langle (\blacktriangleright \cup \blacktriangledown \cup \blacktriangleleft \cup \blacktriangle)^{3} \rangle K_{A} water$$

- A cannot **know** about the goal until the length of the sequence of moves is at least 3.

Are these more efficient fragments?

Star-Free fragment	$[aab+b]K_ip$, $\cancel{[aab^*]K_ip}$	PSPACE-Hard ($TQBF$)
Existential fragment	$\langle aab^*\rangle K_ip$, $\cancel{[aab^*]K_ip}$	PSPACE-Hard (Intersection Non-Emptiness Problem)
Star-Free$-$Existential fragment	$\langle aab+b\rangle \hat{K}_ip$, $\cancel{\langle aab^*\rangle K_ip}$	NP-Complete (3-SAT)
Word fragment	$[aab]K_ip$, $\cancel{[aab+b]K_ip}$	P

POL Satisfiability

Model Checker is a powerful tool, when the scenario is modeled.

POL Satisfiability

Model Checker is a powerful tool, when the scenario is modeled.

Question: What about properties of multiple models?

POL Satisfiability

Model Checker is a powerful tool, when the scenario is modeled.

Question: What about properties of multiple models?

Question: Is a certain property *satisfied* in EVERY model having certain other properties in common?

POL Satisfiability

Model Checker is a powerful tool, when the scenario is modeled.

Question: What about properties of multiple models?

Question: Is a certain property *satisfied* in EVERY model having certain other properties in common?

Approach: Specifying the models using the formulas.

POL Satisfiability

Model Checker is a powerful tool, when the scenario is modeled.

Question: What about properties of multiple models?

Question: Is a certain property *satisfied* in EVERY model having certain other properties in common?

Approach: Specifying the models using the formulas.

For example:

$water \wedge \hat{K}_A power$

POL Satisfiability

Model Checker is a powerful tool, when the scenario is modeled.

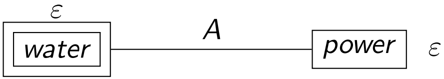
Question: What about properties of multiple models?

Question: Is a certain property *satisfied* in EVERY model having certain other properties in common?

Approach: Specifying the models using the formulas.

For example:

$water \wedge \hat{K}_A power \quad \rightarrow$



,

Model Checker is a powerful tool, when the scenario is modeled.

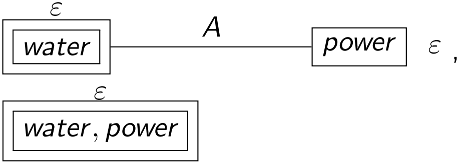
Question: What about properties of multiple models?

Question: Is a certain property *satisfied* in EVERY model having certain other properties in common?

Approach: Specifying the models using the formulas.

For example:

$water \wedge \hat{K}_A power \quad \rightarrow$



POL Satisfiability

Question: Is certain property *satisfied* in EVERY model having certain other properties in common?

POL Satisfiability

Question: Is certain property *satisfied* in EVERY model having certain other properties in common?

Approach: Specifying the models using the formulas : φ_M

POL Satisfiability

Question: Is certain property *satisfied* in EVERY model having certain other properties in common?

Approach: Specifying the models using the formulas : φ_M

Specify the property to verify: φ

POL Satisfiability

Question: Is certain property *satisfied* in EVERY model having certain other properties in common?

Approach: Specifying the models using the formulas : φ_M

Specify the property to verify: φ

Verify whether $\neg(\varphi_M \rightarrow \varphi)$ has a model

POL Satisfiability

Question: Is certain property *satisfied* in EVERY model having certain other properties in common?

Approach: Specifying the models using the formulas : φ_M

Specify the property to verify: φ

Verify whether $\neg(\varphi_M \rightarrow \varphi)$ has a model

If NO MODEL then φ is a property

Else it is not

POL Satisfiability Results

$$\text{POL formula } \varphi \longrightarrow \boxed{\text{POL SAT Checker}} \longrightarrow \exists ? \mathcal{M}, s : \mathcal{M}, s \models \varphi ?$$

POL Satisfiability Results

$$\text{POL formula } \varphi \longrightarrow \boxed{\text{POL SAT Checker}} \longrightarrow \exists ? \mathcal{M}, s : \mathcal{M}, s \vDash \varphi ?$$

Is POL-Sat **decidable**?

POL Satisfiability Results

$$\text{POL formula } \varphi \longrightarrow \boxed{\text{POL SAT Checker}} \longrightarrow \exists ? \mathcal{M}, s : \mathcal{M}, s \models \varphi ?$$

Is POL-Sat **decidable**?
Yes.

POL- Satisfiability Complexity [To be Submitted]

The satisfiability problem of POL is in $\mathsf{DOUBLE-EXPSPACE}$.

POL Satisfiability Results

$$\text{POL formula } \varphi \longrightarrow \boxed{\text{POL SAT Checker}} \longrightarrow \exists ? \mathcal{M}, s : \mathcal{M}, s \vDash \varphi ?$$

Is POL-Sat **decidable**?
Yes.

POL- Satisfiability Complexity [To be Submitted]

The satisfiability problem of POL is in DOUBLE – EXPSPACE.

Open: Lower Bound

POL Fragment Results (KR'23)[1]

Star-Free Multi-agent fragment	$[aab + b]K_i p$, $\cancel{[aab^*]K_i p}$	NEXPTIME-Complete (Tiling Problem)
Star-Free Single-agent fragment	$[aab]K_i p \vee K_i q$, $\cancel{[aab^* + b]K_i p \vee K_j q}$	PSPACE-Hard (TQBF)
Word Multi-agent fragment	$[aab]K_i p$, $\cancel{[aab + b]K_i p}$	PSPACE-Complete (PAL Reduction)
Word Single-agent fragment	$[aab]K_i p \vee K_i q$, $\cancel{[aab + b]K_i p \vee K_j q}$	NP-Complete (PAL Reduction)

[1] Chakraborty S.; Ghosh A.; Ghosh S.; and Schwarzentruber F. 2023. On simple expectations and observations of intelligent agents: A complexity study., KR 2023, Rhodes, Greece, September 2-8, 2023, 136–145.

Table of Contents

Decidability: Model-Checking

- Consider the formula $\langle \pi^{\star} \rangle \psi$.

Decidability: Model-Checking

- Consider the formula $\langle \pi^\star \rangle \psi$.
- As per semantics, given a $\mathcal{M}, s$, we search for a $w \in \mathcal{L}(\pi^\star)$ such that...

Decidability: Model-Checking

- Consider the formula $\langle\pi^\star\rangle\psi$.
- As per semantics, given a $\mathcal{M}, s$, we search for a $w \in \mathcal{L}(\pi^\star)$ such that...
- s survives in $\mathcal{M}|_w$ and...

Decidability: Model-Checking

- Consider the formula $\langle \pi^\star \rangle \psi$.
- As per semantics, given a $\mathcal{M}, s$, we search for a $w \in \mathcal{L}(\pi^\star)$ such that...
- s survives in $\mathcal{M}|_w$ and...
- Recursively check $\mathcal{M}|_w, s \vDash \psi$.

Decidability: Model-Checking

- Consider the formula $\langle \pi^\star \rangle \psi$.
- As per semantics, given a $\mathcal{M}, s$, we search for a $w \in \mathcal{L}(\pi^\star)$ such that...
- s survives in $\mathcal{M}|_w$ and...
- Recursively check $\mathcal{M}|_w, s \vDash \psi$.
- **Problem:** How far in $\mathcal{L}(\pi^\star)$ to search since the language will have infinite words?

Decidability: Model-Checking

- Consider the formula $\langle \pi^\star \rangle \psi$.
- As per semantics, given a $\mathcal{M}, s$, we search for a $w \in \mathcal{L}(\pi^\star)$ such that...
- s survives in $\mathcal{M}|_w$ and...
- Recursively check $\mathcal{M}|_w, s \vDash \psi$.
- **Problem:** How far in $\mathcal{L}(\pi^\star)$ to search since the language will have infinite words?
- **Solution:** $\pi^\star$ is a regular expression $\longrightarrow$ NFA/DFA (**FINITE**)

Decidability: Model-Checking

- Consider the formula $\langle \pi^\star \rangle \psi$.
- As per semantics, given a $\mathcal{M}, s$, we search for a $w \in \mathcal{L}(\pi^\star)$ such that...
- s survives in $\mathcal{M}|_w$ and...
- Recursively check $\mathcal{M}|_w, s \vDash \psi$.
- **Problem:** How far in $\mathcal{L}(\pi^\star)$ to search since the language will have infinite words?
- **Solution:** $\pi^\star$ is a regular expression $\longrightarrow$ NFA/DFA (**FINITE**)
- Given a model $\mathcal{M}$, bound the number of **unique** $\mathcal{M}|_w$ over any $w \in \Sigma^\star$.

Decidability of POL^{-}: Tableau Term

- **Survival Term**

Decidability of POL^{-}: Tableau Term

- **Survival Term**
  - $(\sigma \quad w \quad \checkmark)$: World σ survives after updated by w.

Decidability of POL^{-}: Tableau Term

- **Survival Term**
  - $(\sigma \quad w \quad \checkmark)$: World σ survives after updated by w.
- **Formula Term**

Decidability of POL$^-$: Tableau Term

- **Survival Term**
  - $(\sigma \quad w \quad \checkmark)$: World σ survives after updated by w.

- **Formula Term**
  - $(\sigma \quad w \quad \varphi)$: φ holds in World σ after updating by w

Decidability of POL^{-}: Some Tableau Rules

Propositional Rules

Clash rule
$$\frac{(\sigma \quad w \quad p), \quad (\sigma \quad w \quad \neg p)}{\perp}$$

Diamond and Box Rules

Diamond Project
$$\frac{(\sigma \quad w \quad \langle a \rangle \psi)}{(\sigma \quad wa \quad \checkmark), (\sigma \quad wa \quad \psi)}$$

Box Project
$$\frac{(\sigma \quad w \quad [\pi]\psi), \quad (\sigma \quad wa \quad \checkmark)}{(\sigma \quad wa \quad [\pi \backslash a]\psi)}$$

Survival Rules

Constant Valuation Up
$$\frac{(\sigma \quad w \quad p)}{(\sigma \quad \epsilon \quad p)} \qquad \frac{(\sigma \quad w \quad \neg p)}{(\sigma \quad \epsilon \quad \neg p)}$$

Survival Chain
$$\frac{(\sigma \quad wa \quad \checkmark)}{(\sigma \quad w \quad \checkmark)}$$

Decidability: Satisfiability

Again consider the formula: $\langle \pi^\star \rangle \psi$.

Decidability: Satisfiability

Again consider the formula: $\langle\pi^\star\rangle\psi$.

Say there is a model $\mathcal{M}, s \vDash \langle\pi^\star\rangle\psi$

Decidability: Satisfiability

Again consider the formula: $\langle \pi^\star \rangle \psi$.

Say there is a model $\mathcal{M}, s \vDash \langle \pi^\star \rangle \psi$

This implies either $\mathcal{M}, s \vDash \psi$ (Inductively checked), where $\epsilon \in \mathcal{L}(\pi^\star)$ is the witness word.

Decidability: Satisfiability

Again consider the formula: $\langle \pi^\star \rangle \psi$.

Say there is a model $\mathcal{M}, s \vDash \langle \pi^\star \rangle \psi$

This implies either $\mathcal{M}, s \vDash \psi$ (Inductively checked), where $\epsilon \in \mathcal{L}(\pi^\star)$ is the witness word.

OR $\mathcal{M}, s \vDash \langle \pi \rangle \langle \pi^\star \rangle \psi$

Decidability: Satisfiability

Again consider the formula: $\langle \pi^\star \rangle \psi$.

Say there is a model $\mathcal{M}, s \vDash \langle \pi^\star \rangle \psi$

This implies either $\mathcal{M}, s \vDash \psi$ (Inductively checked), where $\epsilon \in \mathcal{L}(\pi^\star)$ is the witness word.

OR $\mathcal{M}, s \vDash \langle \pi \rangle \langle \pi^\star \rangle \psi$

The input formula becomes subformula of a bigger formula $=>$ problem in proving inductively.

Decidability: Satisfiability

Again consider the formula: $\langle \pi^\star \rangle \psi$.

Say there is a model $\mathcal{M}, s \models \langle \pi^\star \rangle \psi$

This implies either $\mathcal{M}, s \models \psi$ (Inductively checked), where $\epsilon \in \mathcal{L}(\pi^\star)$ is the witness word.

OR $\mathcal{M}, s \models \langle \pi \rangle \langle \pi^\star \rangle \psi$

The input formula becomes subformula of a bigger formula $=>$ problem in proving inductively.

Approach: Look towards Automaton structures

Table of Contents

Concluding...

- Model-Checking and Satisfiability problem for POL (Full language ongoing)
- Complete Axiomatic System for extension of POL : Epistemic Protocol Logic[1]
- Programs can be interpreted more efficiently in CFL.
  How about CFG instead of regular?

[1] H van Ditmarsch, S Ghosh, R Verbrugge, and Y Wang. Hidden protocols: Modifying our expectations in an evolving world. Artificial Intelligence, 208:18–40, 2014