Intelligent Agents

Knowledge-Based Programs
(for decentralized POMDPs)

Özgür L. Özçep
Universität zu Lübeck
Institut für Informationssysteme

Todays lecture based on

- The AAMAS 2019 Tutorial **„EPISTEMIC REASONING IN MULTI-AGENT SYSTEMS“**
  http://people.irisa.fr/Francois.Schwarzentruber/2019AAMAStutorial/

MOTIVATION

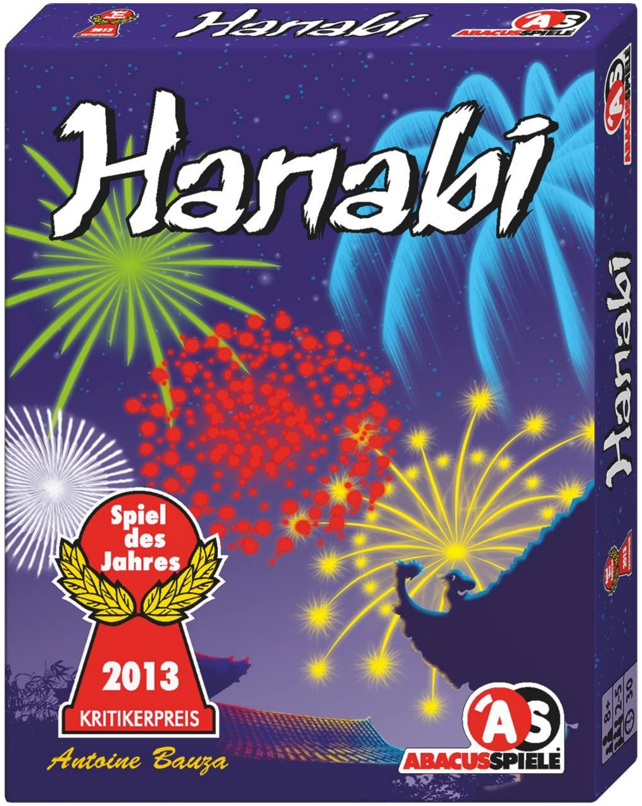
Setting

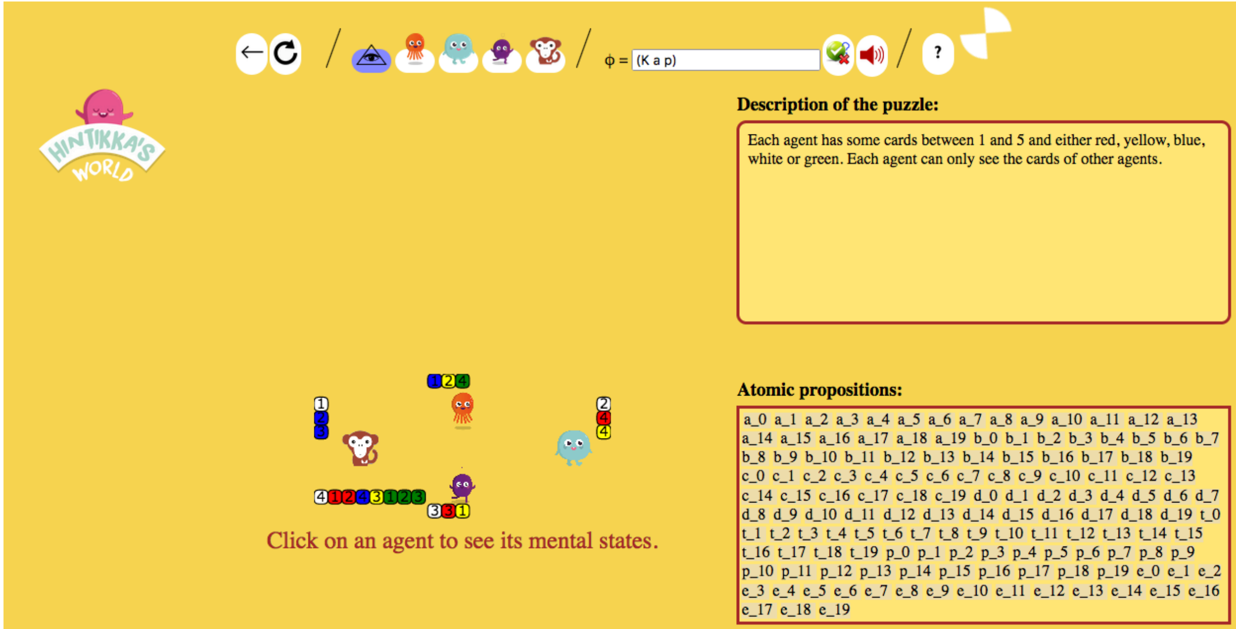
- Cooperative agents: in particular, common (offline) joint plan
- Common goal
- Imperfect information
- Decentralized execution by agents

- QDec-POMDPs: (qualitative) Decentralized Partially Observable Markov Decision Problems (Brafman et al 13)

- Main observation: Policies are large

Similar setting ...





Methodology

- Planning ...
  - Input
    - Model
    - Goal
  - Output
    - Program for each agent involved
- ... That is undestandable (explainable)
  - Legal issues in case of failure
  - Interaction with humans (-> human-compatible AI)
  - Pointers to lines in code that caused error may help but isfar from explainability in a strict sense
- ... And that is succinct (policies tend to be large)

Running Example (Saffidine et al. 18)

„...assume Alice wants to meet Bob. She may take a train or a flight from her place to his, but there is a risk of air crew strike. Alice and Bob may call each other beforehand (centralized planning phase) to agree on the following plan.
Alice tries to fly. If there is a strike, she takes the train and listens to the radio to know whether Bob can know this. If the strike is not announced on the radio (hence Bob cannot know), after arriving at the station she will reach the airport to meet Bob there. Otherwise they will meet at her place of arrival. It can be seen that this plan, once agreed upon, can be executed successfully in a decentralized fashion.

Using knowledge-based programs (KBPs)

KBP for agent a

listenToRadio

If a knows strike

then toStation

else toAirport

KBP for agent b

readNewsPaper

If b knows strike

then **toStation**

else toAirport

- Operational semantics
- Succinctness
- (Un)decidability/complexity results

- More information to be found in the papers
  (Lang/ Zanuttini 12; 13), (Zanuttini et al. 20), (Seffadine et al 18)

SYNTAX AND SEMANTICS

Knowlede-Based Programs: Syntax

See (Fagin et al. 03) for an in-depth treatment of KBPs

Definition (Syntax of KBPs)

- Actions: turn left, stay, broadcast temperature etc.
- Sequence: ... ; ....
- **if** ϕ **then** ... **else** ...
- **while** ϕ **do** ...

Example (KBP for agent a)

If a knows (door 12 is locked and justobserved(fire)) **then**
turn left
broadcast temperature
Else
stay

Resctricton on KBPs

- Each KBP is a KBP(i) for some agent i
- The formulae ϕ used in the conditions of if-then and while in KBP(i) are epistemic formuale that are subjective for i (all propositions are in scope of K_i modality)

- ϕ may also contain assertions of the form justObserved(x) expressing the fact that x is the last observation

Running Example (Saffidine et al. 18)

KBP for agent Alice a

try-plane

If $K_a\ (\neg\ plane_a)$

then

take-train;

turn-radio-on:

listen-radio;

If $K_a(\neg K_b strike)$

then to-airport

KBP for agent Bob b

turn-radio-on;

listen-radio;

If $K_b(strike)$

then to-station

else to-airport

Semantics

Definition

- QdecPomDP = Qualitative decentralized partially Observable Markov Decision Problem
  = Concurrent game structures with observations

- Transitions of the form

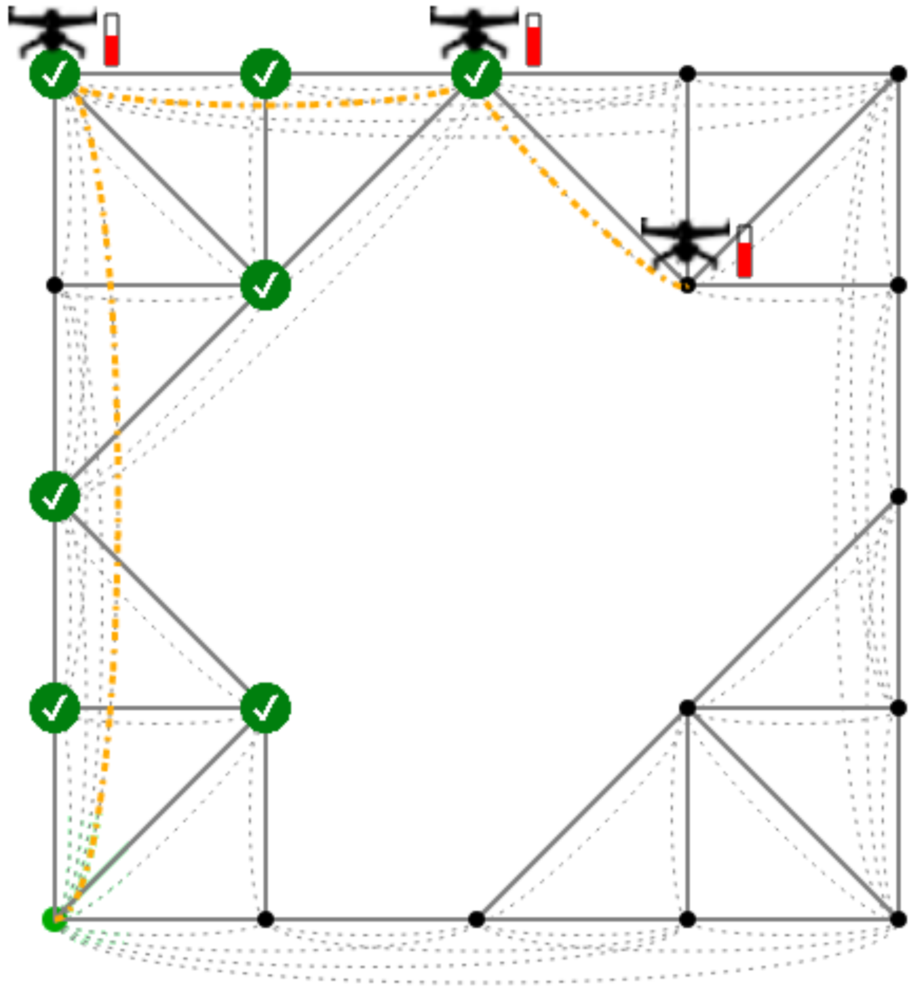
a: stay | b: turn left | a: observes fire | b: observes ashes

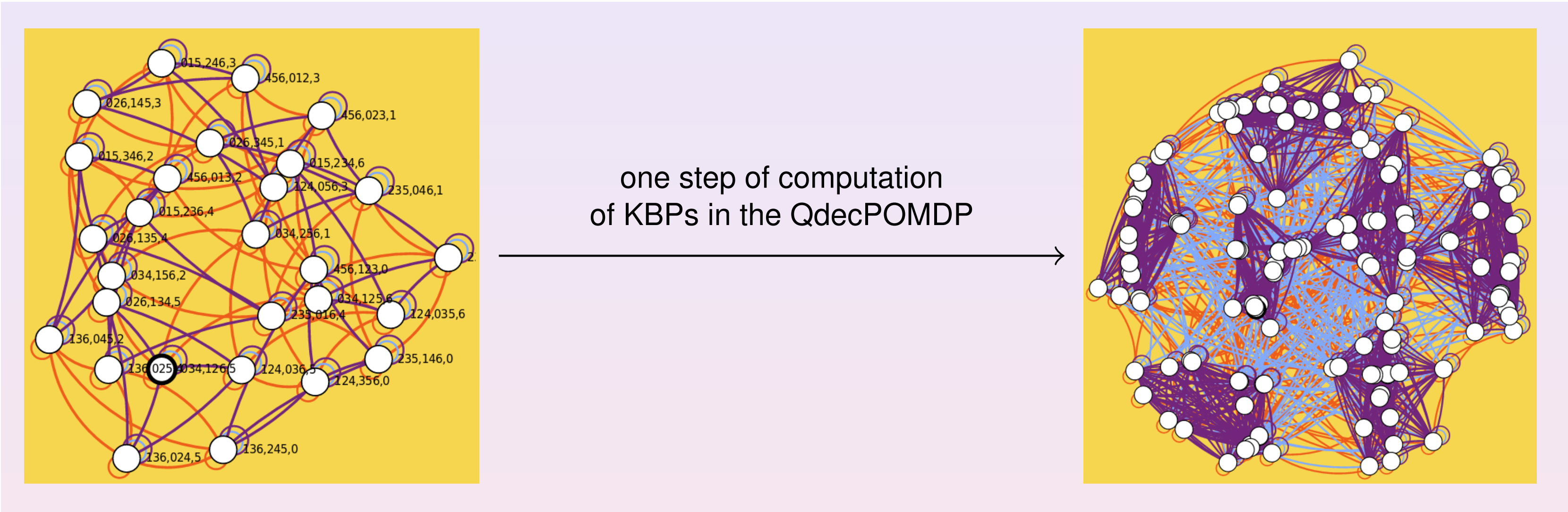
state1 ⟶ state2

- Non-empty set of possible initial states
- Set of goal states

Typically, a state describes

- Position of agents
- Battery levels
- Etc.





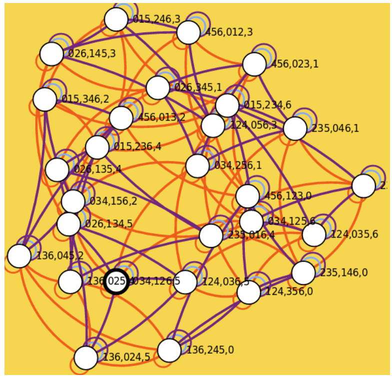
Epistemic structure: Higher-order knowledge about:

- The current state of the QdecPOMDP
- The current program counters in KBPs

Assumptions

- Common knowledge of
  - The QdecPOMDP
  - The KBPs
  - Synchrony of the system
    - Tests last 0 unit of time
    - Actions last 1 unit of time

Epistemic Structures at time T: worlds



- Worlds = consistent histories of the form

$$s^0\ \overrightarrow{pc}^0\ \overrightarrow{obs}^1 s^1\ \overrightarrow{pc}^1 \ldots \overrightarrow{obs}^T s^T\ \overrightarrow{pc}^T$$

  where

  - $\overrightarrow{obs}^1$: vector of observations at time t
  - s^t: state at time t
  - $\overrightarrow{pc}^t$: vector of program counters at time t

listenToRadio

If K_a strike

toStation

else toAirport

Epistemic struct.: Indistinguishability relations

- Two histories are indistinguishable for an agent a iff she has received the same observations

- Formally:

$$s^0\ \overrightarrow{pc}^0\ \overrightarrow{obs}^1 s^1\ \overrightarrow{pc}^1 \ldots \overrightarrow{obs}^T s^T\ \overrightarrow{pc}^T \quad \sim_a$$
$$s'^0\ \overrightarrow{pc}'^0\ \overrightarrow{obs}'^1 s'^1\ \overrightarrow{pc}'^1 \ldots \overrightarrow{obs}'^T s'^T\ \overrightarrow{pc}'^T$$

iff

for all $t \in \{1, \ldots, T\}$: $\overrightarrow{obs_a}^t = \overrightarrow{obs_a}'^t$

Motivating Program Counters (PCs)

- Assume, Agent i has to evaluate: If $K_i \mathrm{K_j}\ \phi$
- Requires reasoning on observations j has collected so far;
- due to partial observability i may not know about actions taken so far by j
- To cope with this introduce PCs: (Γ, a, κ)
  - Intended meaning:
  - If all formulae in Γ are currently true j is about to execute action a, then it will execute KBP κ

Definition (Program counter)

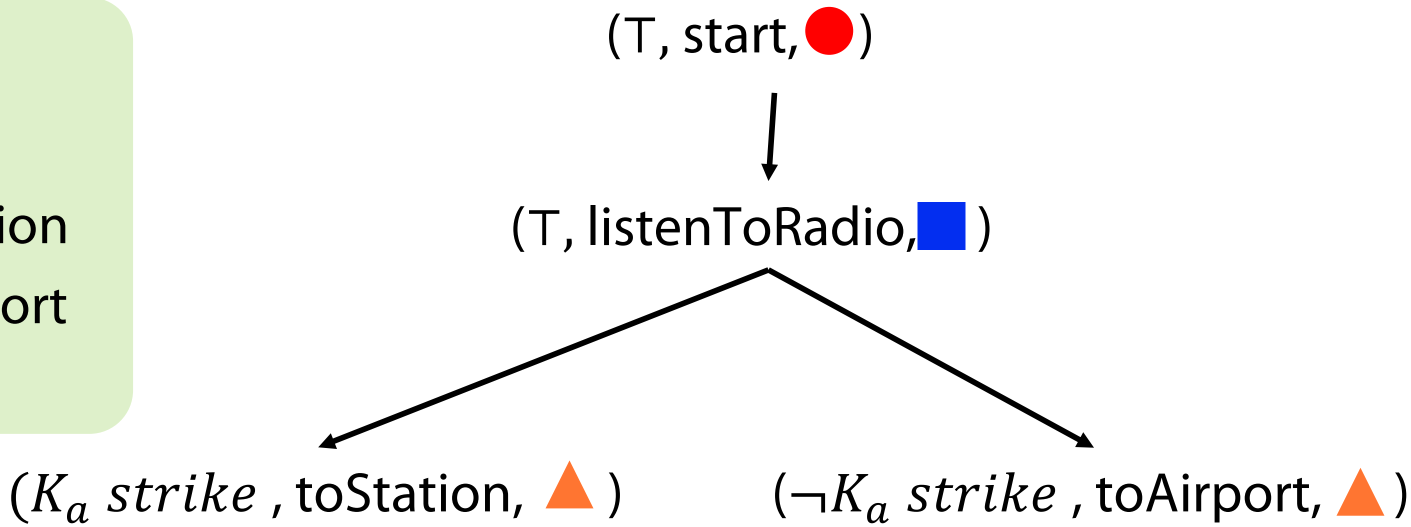
(guard, action just executed, continuation)

● listenToRadio
■ **If** $\boldsymbol{K_a}$ strike
  toStation
else toAirport
▲

- ($\top$, start, ●)
- ($\top$, listenToRadio, ■)
- ($K_a\ strike$, toStation, ▲)
- ($\neg K_a\ strike$, toAirport, ▲)

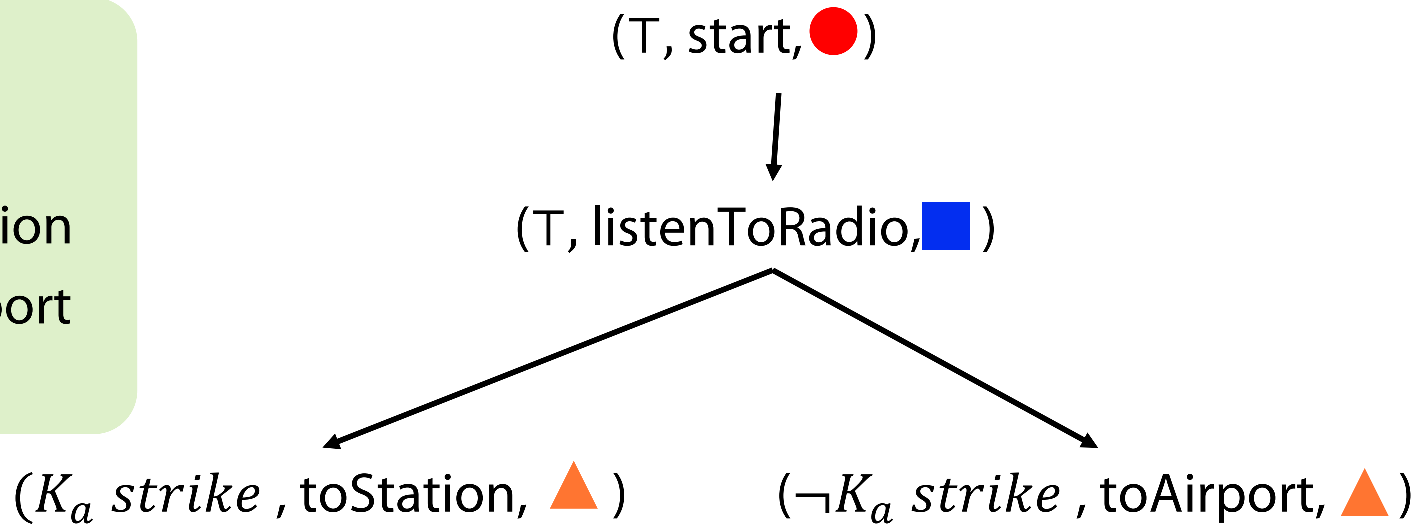
Control flow graph

listenToRadio
If $\boldsymbol{K_a}$ strike
toStation
else toAirport

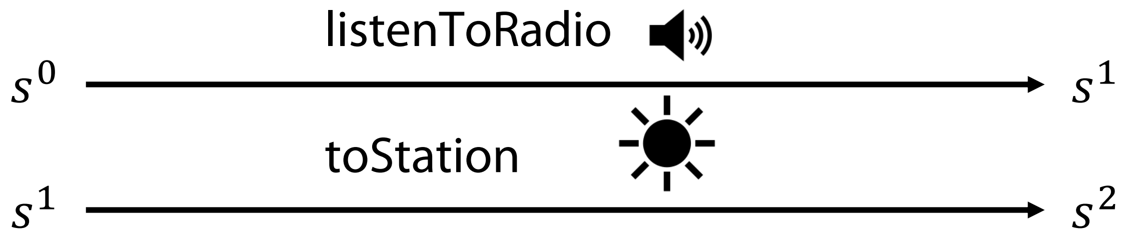


Control flow graph

- listenToRadio
- If K_a strike
  - toStation
- else toAirport



In the QdecPOMDP:



Consistent history:

s^0(⊤, start,); s^1(⊤, listenToRadio,); s^2(⊤, K_a $strike$, toStation,)

$\models K_a$ $strike$

Connection to dynamic logic

Structure at time t given by

- (iterated) product update
- starting with situation encoding common knowledge of initial belief state,
- and event model built from the control-flow graphs of the KBPs.

ALGORITHMIC PROBLEMS

Verification problem

- Input
  - A QDecPOMDP model (given in STRIPS-like symbolic form);
  - Knowledge-based programs for each agent

- Output: yes if all executions of the KBPs lead to a goal state.

Theorem

The verification problem for while-free KBPs is PSPACE-complete

- Proof idea
  - Upper bound: on-the-fly model checking
  - Lower bound: reduction from TQBF (true quantified Boolean formulae)

Execution Problem

- Input
  - An agent a
  - A QdecPOMDP model
  - Policies (e.g. KBPs), one for each agent
  - A local view of the histoty for agent a.
- Output: the action act agent a should take

- Note
  - Problem is trivial for other representations such as joint policy trees (reactive representations, next slide).
  - But succinctness of KBPs requires (more) reasoning in execution of the plan.

Execution problem

Definition

A class of policy representations is reactive iff
its corresponding execution problem is in P

Example

Tree policies are reactive policy representations

If justobserved(fire) **then** turn left **else** stay

Unless P = PSPACE, KBPs are not reactive. Indeed:

Proposition

The execution problem for KBPs is PSPACE-complete

Succinctness

- Remember the notion of modal depth

Definition

The size/length and the modal depth of formulae are defined as follows:

- $|p| = 1$ $\quad d(p) = 0$
- $|\neg \phi| = |\phi| + 1$ $\quad d(\neg\phi) = d(\phi)$
- $|\phi \wedge \psi| = |\phi| + |\psi| + 1$ $\quad d(\phi \wedge \psi) = \max\{d(\phi), d(\psi)\}$
- $|K_a \phi| = \phi + 1$ $\quad d(K_a \phi) = 1 + d(\phi)$

Example

$$d(justObserved(fire)) = 0$$
$$d(K_a\, p) = 1$$
$$d\big(K_a\, (K_b\, p)\big) = 2$$

Succinctness

Theorem (Lang/Zanuttini 12 for d = 1; Saffidine et al 18, for d > 1)

Let $d \geq 1$. There is a poly(n)-size QdecPOMDP family $\left(M_{n,d}\right)_{n\in\mathbb{N}}$

For which:

1. There is a d-modal depth poly(n)-size valid KBP family
2. No (d-1)-modal depth valid KBP family
3. Assuming NP $\not\subseteq$ P/poly
   for any reactive policy representations, no poly(n)-size valid policy family

(Complexity theory reminder)

Two (equivalent) definitions of P/Poly

A language L is in P/poly iff

1. „there is a language A in P and a set of advice strings $\{a_0,a_1,\ldots\}$ such that $|a_n|\leq n^{O(1)}$ and x is in L if and only if $(x,a_{|x|})$ is in A.
2. there is a family of circuits $\{C_0,C_1,\ldots\}$ such that $|C_n|\leq n^{O(1)}$ and for all n and all $x=x_1\ldots x_n$, x is in L if and only if $C_n(x_1,\ldots,x_n)$ accepts."

(http://blog.computationalcomplexity.org/2005/09/ppoly.html)

CONCLUSION

Methodology

- Planning ...
  - Input
    - Model
    - Goal
  - Output
    - Program for each agent involved ->reactive policy for each agent

- Higher-order knowledge for
  - Getting explainable policies (e.g. for making cooperation visible)
  - For concise programs

Perspectives

- Efficient implementation of the verification/execution problems;
- Heuristics for the planning problem;
- More tractable fragments;
- decPOMDP (with probabilities);
- Temporal properties;
- Strategic reasoning;
- Develop proof systems for KBPs. Use of Coq, Isabelle?

Uhhh, a lecture with a hopefully useful

APPENDIX

References

- **(Brafman et al 13)**
  [1] R. I. Brafman, G. Shani, and S. Zilberstein. Qualitative planning under partial observability in multi- agent domains. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI'13, pages 130–137. AAAI Press, 2013.
- **(Lang/Zanuttini 12)**
  J. Lang and B. Zanuttini. Knowledge-based programs as plans - the complexity of plan verification. In L. D. Raedt, C. Bessiere, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. J. F. Lucas, editors, ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012, volume 242 of Frontiers in Artificial Intelligence and Applications, pages 504–509. IOS Press, 2012.
- **(Lang/Zanuttini 13)**
  J. Lang and B. Zanuttini. Knowledge-based programs as plans: Succinctness and the complexity of plan existence. In B. C. Schipper, editor, Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2013), Chennai, India, January 7-9, 2013, 2013.
- **(Zannutini et al. 20)**
  B. Zanuttini, J. Lang, A. Saffidine, and F. Schwarzentruber. Knowledge-based programs as succinct policies for partially observable domains. Artificial Intelligence, 288(C):103365, 2020.
- **(Fagin et al. 03)**
  R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. Reasoning about Knowledge. MIT Press, 2003.
- **(Saffidine et al 18)**
  A. Saffidine, F. Schwarzentruber, and B. Zanuttini. Knowledge-based policies for qualitative decen- tralized pomdps. In S. A. McIlraith and K. Q. Weinberger, editors, Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 6270–6277. AAAI Press, 2018.

Color Convention in this course

- Formulae, when occurring inline
- Newly introduced terminology and definitions
- Important results (observations, theorems) as well as emphasizing some aspects
- Examples are given with standard orange with possibly light orange frame
- Comments and notes
- Algorithms