Intelligent Agents
Knowledge-Based Programs
(for decentralized POMDPs)

Özgür L. Özçep
Universität zu Lübeck
Institut für Informationssysteme
Today's lecture based on

- The AAMAS 2019 Tutorial "EPISTEMIC REASONING IN MULTI-AGENT SYSTEMS"
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 ...

Description of the puzzle:
Each agent has some cards between 1 and 5 and either red, yellow, blue, white or green. Each agent can only see the cards of other agents.

Atomic propositions:

\[
\begin{align*}
& \text{a}_0, \text{a}_1, \text{a}_2, \text{a}_3, \text{a}_4, \text{a}_5, \text{a}_6, \text{a}_7, \text{a}_8, \text{a}_9, \text{a}_{10}, \text{a}_{11}, \text{a}_{12}, \text{a}_{13}, \\
& \text{b}_0, \text{b}_1, \text{b}_2, \text{b}_3, \text{b}_4, \text{b}_5, \text{b}_6, \text{b}_7, \\
& \text{c}_0, \text{c}_1, \text{c}_2, \text{c}_3, \text{c}_4, \text{c}_5, \text{c}_6, \text{c}_7, \text{c}_8, \text{c}_9, \text{c}_{10}, \text{c}_{11}, \text{c}_{12}, \text{c}_{13}, \\
& \text{d}_0, \text{d}_1, \text{d}_2, \text{d}_3, \text{d}_4, \text{d}_5, \text{d}_6, \text{d}_7, \\
& \text{e}_0, \text{e}_1, \text{e}_2, \text{e}_3, \text{e}_4, \text{e}_5, \text{e}_6, \text{e}_7, \text{e}_8, \text{e}_9, \text{e}_{10}, \text{e}_{11}, \text{e}_{12}, \text{e}_{13}, \text{e}_{14}, \text{e}_{15}, \text{e}_{16} \\
& \text{p}_0, \text{p}_1, \text{p}_2, \text{p}_3, \text{p}_4, \text{p}_5, \text{p}_6, \text{p}_7, \text{p}_8, \text{p}_9, \\
& \text{c}_0, \text{c}_1, \text{c}_2, \text{c}_3, \text{c}_4, \text{c}_5, \text{c}_6, \text{c}_7, \text{c}_8, \text{c}_9, \text{c}_{10}, \text{c}_{11}, \text{c}_{12}, \text{c}_{13}, \text{c}_{14}, \text{c}_{15}, \text{c}_{16}, \\
& \text{e}_{17}, \text{e}_{18}, \text{e}_{19}, \text{e}_{20}, \text{e}_{21}, \text{e}_{22}, \text{e}_{23}, \text{e}_{24}, \text{e}_{25}, \text{e}_{26}, \text{e}_{27}, \text{e}_{28}, \text{e}_{29}, \text{e}_{30}, \text{e}_{31}, \text{e}_{32}, \text{e}_{33}, \text{e}_{34}, \text{e}_{35}, \text{e}_{36}, \text{e}_{37}, \text{e}_{38}, \text{e}_{39} \\
\end{align*}
\]
Methodology

• Planning ...
  – Input
    • Model
    • Goal
  – Output
    • Program for each agent involved
• ... That is understandable (explainable)
  – Legal issues in case of failure
  – Interaction with humans (-> human-compatible AI)
  – Pointers to lines in code that caused error may help but is far 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 \( \phi \) then ... else ...
- while \( \phi \) do ...

Example (KBP for agent a)

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

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

- $\phi$ 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 $K_a (\neg \text{plane}_a)$
then
  take-train;
  turn-radio-on:
  listen-radio;
  If $K_a (\neg K_b \text{strike})$
then to-airport

KBP for agent Bob b

turn-radio-on;
listen-radio;
If $K_b (\text{strike})$
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

- 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
  \[ s^0 \overrightarrow{pc^0} obs^1 s^1 \overrightarrow{pc^1} \ldots obs^T s^T \overrightarrow{pc^T} \]

  where
  - \( obs^1 \): vector of observations at time \( t \)
  - \( s^t \): state at time \( t \)
  - \( pc^t \): vector of program counters at time \( t \)
Epistemic struct.: Indistinguishability relations

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

- Formally:

  $$
  s^0 \overset{pc}{\rightarrow}^0 obs^1 s^1 \overset{pc}{\rightarrow}^1 ... obs^T s^T \overset{pc}{\rightarrow}^T \sim_a
  s'^0 \overset{pc'}{\rightarrow}^0 obs'^1 s'^1 \overset{pc'}{\rightarrow}^1 ... obs'^T s'^T \overset{pc'}{\rightarrow}^T
  $$

  iff

  for all $t \in \{1, ..., T\}$: $\overset{t}{obs}_a = \overset{t}{obs'}_a$
Motivating Program Counters (PCs)

- Assume, Agent i has to evaluate: If $K_i K_j \phi$
- 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: $(\Gamma, a, \kappa)$
  - Intended meaning:
    - If all formulae in $\Gamma$ are currently true $j$ is about to execute action $a$, then it will execute KBP $\kappa$
Definition (Program counter)

(guard, action just executed, continuation)

- \((\top, \text{start}, \bullet)\)
- \((\top, \text{listenToRadio}, \square)\)
- \((K_a \text{ strike}, \text{toStation}, \triangleright)\)
- \((\neg K_a \text{ strike}, \text{toAirport}, \triangle)\)
Control flow graph

- If $K_a \text{ strike}$
  - toStation
- else
  - toAirport

- (T, start, red dot)

- (T, listenToRadio, blue square)

- ($K_a \text{ strike}$, toStation, orange triangle)

- ($\neg K_a \text{ strike}$, toAirport, orange triangle)
Control flow graph

In the QdecPOMDP:

Consistent history:

\[ s^0(T, \text{start, } \bullet); \quad \text{listenToRadio}; \quad s^1(T, \text{listenToRadio, } \blacksquare); \quad s^2(T, K_a \text{ strike, toStation, } \triangle) \subseteq K_a \text{ strike} \]
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.
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 history 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.
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:

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

**Example**

$\quad d(\text{justObserved(fire)}) = 0$
$\quad d(K_a p) = 1$
$\quad d(K_a (K_b p)) = 2$
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 $\text{QdecPOMDP}$ family $(M_{n,d})_{n \in \mathbb{N}}$ 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?
Uhuh, a lecture with a hopefully useful

APPENDIX
References

- **(Brafman et al 13)**

- **(Lang/Zanuttini 12)**

- **(Lang/Zanuttini 13)**

- **(Zannutini et al. 20)**

- **(Fagin et al. 03)**

- **(Saffidine et al 18)**
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