Knowledge-Based Policies for Qualitative Decentralized POMDPs

Abdallah Saffidine   Bruno Zanuttini
François Schwarzentruber

68NQRT

January 25th, 2018
Automation of complex tasks

Building surveillance  Nuclear decommissioning  Intelligent farming
Multiple robots

more robust/efficient than
Multiple robots

more robust/efficient than
Multiple robots

more robust/efficient than
Multiple robots

more robust/efficient than

Settings

- Cooperative agents;
- Common goal;
- Imperfect information;
- Decentralized execution.
Methodology
Need: understandable system

Motivation

- Legal issues in case of failure
- Interaction with humans

```c
#include "fixed.h"
#include "fixed_private.h"

int16_t error;
int16_t torque_request;
DWork DWork;

void fixed_step(void)
{
    int16_t FilterCoefficient_m;
    FilterCoefficient_m = (int16_t)((int32_t)((int16_t)(5403L * (int32_t)error >>
        1U) - DWork.Filter_DSTATE) << 4U) * 17893L >> 14);
    torque_request = ((int16_t)(12475L * (int32_t)error >> 14U) >> 1) +
        (DWork.Integrator_DSTATE >> 2) - (FilterCoefficient_m >> 1);
    DWork.Integrator_DSTATE = (int16_t)((int32_t)error >> 13U) + 5242L >>
        1U) + DWork.Integrator_DSTATE;
    DWork.Filter_DSTATE = (int16_t)(5242L * (int32_t)FilterCoefficient_m >> 16U) +
        DWork.Filter_DSTATE;
}

void fixed_initialize(void)
{
    torque_request = 0;

    (void) memset((void *)&DWork, 0,
        sizeof(DWork));
    error = 0;
}
```
Our contribution: use of knowledge-based programs

KBP for agent a

listenRadio

if a knows strike

| toStation

else

| toAirport

KBP for agent b

readNewsPaper

if b knows strike

| toStation

else

| toAirport

- Operational Semantics for Knowledge-based programs;
- (Un)decidability/complexity and succinctness.

Extends: [Lang, Zanuttini, ECAI2012, TARK2013]
Outline

1. Knowledge-based programs
   - Epistemic formulas
   - Program constructions

2. Semantics

3. Mathematical properties

4. Conclusion
Outline

1. Knowledge-based programs
   - Epistemic formulas
   - Program constructions

2. Semantics

3. Mathematical properties

4. Conclusion
Properties expressed in epistemic logic

**Language constructions**

- room 43 is safe
- door 12 is locked
- ... not ...
- (... or ...)  
- (... and ...)  
- (... → ...)  
- (... knows ...)  
- (... knowswhether ...)  

**Example**

(a knows door 12 is locked) and not (c knows door 12 is locked)  

a knowswhether (c knows door 12 is locked)
Outline

1. Knowledge-based programs
   - Epistemic formulas
   - Program constructions

2. Semantics

3. Mathematical properties

4. Conclusion
Program constructions

### Language constructions

- turn left
- stay
- broadcast temperature

- \( \ldots; \ldots \)
- \textbf{if} \ \varphi \ \textbf{then} \ \ldots \ \textbf{else} \ \ldots \\
- \textbf{while} \ \varphi \ \textbf{do} \ \ldots

### Example (knowledge-based program for agent \( a \))

\[
\textbf{if } a \ \textbf{knows} \ (\text{door 12 is locked} \ \text{and} \ \text{justobserved(\text{\textcolor{orange}{\#}})}) \ \textbf{then} \\
\quad \text{turn left} \\
\quad \text{broadcast temperature} \\\n\textbf{else} \\
\quad \text{stay}
\]
Outline

1 Knowledge-based programs

2 Semantics
   - Models: QdecPOMDP
   - Interlude: semantics of epistemic formulas
   - Operational semantics of KBPs

3 Mathematical properties

4 Conclusion
Outline

1. Knowledge-based programs

2. Semantics
   - Models: QdecPOMDP
   - Interlude: semantics of epistemic formulas
   - Operational semantics of KBPs

3. Mathematical properties

4. Conclusion
QdecPOMDP

Qualitative decentralized Partially Observable Markov Decision Processes
= Concurrent game structures with observations.

Transitions of the form:

\[ \begin{align*}
    a & : \text{stay} & a & : \text{fire} \\
    b & : \text{turn left} & b & : \text{snow} \\
\end{align*} \]

state1 \[\rightarrow\] state2

A non-empty set of possible initial states;
A set of goal states.
States

Typically, a state describes:
- positions of agents;
- battery levels;
- etc.
Outline

1. Knowledge-based programs

2. Semantics
   - Models: QdecPOMDP
   - Interlude: semantics of epistemic formulas
   - Operational semantics of KBPs

3. Mathematical properties

4. Conclusion
Prototype

http://people.irisa.fr/Francois.Schwarzentruber/hintikkasworld/
Semantics of epistemic formulas

Epistemic structure $S, w$

$S, w \models a \text{ knows } \varphi$ iff for all $u$, $w \sim_a u$ implies $S, u \models \varphi$. 
Outline

1. Knowledge-based programs

2. Semantics
   - Models: QdecPOMDP
   - Interlude: semantics of epistemic formulas
   - Operational semantics of KBPs

3. Mathematical properties

4. Conclusion
Operational semantics

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;
- synchronicity of the system;
  - tests last 0 unit of time;
  - actions last 1 unit of time.

KBP for agent $a$

- listenRadio

  if $a$ knows $strike$
  
  toStation

  else
  
  toAirport

KBP for agent $b$

- readNewsPaper

  if $b$ knows $strike$
  
  toStation

  else
  
  toAirport
Epistemic structures at time $T$: worlds

$$\text{Worlds} = \text{consistent histories of the form}$$

$$(\text{wait few slides})$$

$$s_0 \overrightarrow{pc}_0 \overrightarrow{obs}_1 s_1 \overrightarrow{pc}_1 \ldots \overrightarrow{obs}_T s_T \overrightarrow{pc}_T$$

where

<table>
<thead>
<tr>
<th>$\overrightarrow{obs}_t$</th>
<th>vector of observations at time $t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_t$</td>
<td>state at time $t$</td>
</tr>
<tr>
<td>$\overrightarrow{pc}_t$</td>
<td>vector of program counters at time $t$</td>
</tr>
</tbody>
</table>
Epistemic structures at time $t$: indistinguishability relations

Agent $a$ confuses two histories iff she has received the same observations.

\[
s^0 \xrightarrow{p_c^0} \overrightarrow{obs}^1 s^1 \xrightarrow{p_c^1} \cdots \xrightarrow{obs^T} s'^T \xrightarrow{p_c'T}
\]

\[
s'0 \xrightarrow{p_c'0} \overrightarrow{obs}'^1 s'^1 \xrightarrow{p_c'1} \cdots \xrightarrow{obs'T} s'\overrightarrow{T} \xrightarrow{p_c'T}
\]

iff for all $t \in \{1, \ldots, T\}$,

\[
\overrightarrow{obs}_a^t = \overrightarrow{obs}'_a^t
\]
Program counters

Definition (Program counter)
(guard, action just executed, continuation)

- red circle: listenRadio
- green rectangle: if $K_a strike$ then
  - toStation
- yellow triangle: else
  - toAirport

- $\top$, start, ●
- $\top$, listenRadio, □
- $K_a strike$, toStation, ▲
- $\neg K_a strike$, toAirport, ▲
Control-flow graph

- **listenRadio**
- **if** $K_a \text{strike} \text{ then}$
  - toStation
- **else**
  - toAirport

$$(T, \text{start}, \bullet)$$

$$(T, \text{listenRadio}, \square)$$

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

$$(\neg K_a \text{strike}, \text{toAirport}, \triangle)$$
Consistent histories (explained with one agent)

In the QdecPOMDP:

\[ s^0 \xrightarrow{\text{listenRadio}} s^1 \]
\[ s^1 \xrightarrow{\text{toStation}} s^2 \]

KBP control-flow graph

\[ (T, \text{start}, \bullet) \]

\[ (T, \text{listenRadio}, \blacksquare) \]

\[ (K_a \text{strike}, \text{toStation}, \triangle) \]
\[ (\neg K_a \text{strike}, \text{toAirport}, \triangle) \]

\[ s^0 (T, \text{start}, \bullet) \xrightarrow{\text{listenRadio}} s^1 (T, \text{listenRadio}, \blacksquare) \xrightarrow{\text{toStation}} s^2 (K_a \text{strike}, \text{toStation}, \triangle) \]

\[ \models K_a \text{strike} \]
Outline

1. Knowledge-based programs
2. Semantics
3. Mathematical properties
   - Verification
   - Execution problem
   - Succinctness
4. Conclusion
Outline

1. Knowledge-based programs
2. Semantics
3. Mathematical properties
   - Verification
   - Execution problem
   - Succinctness
4. Conclusion
Verification problem

**Input:**
- A QdecPOMDP model;
- Knowledge-based programs for each agent;

**Output:** yes if all executions of the KBPs lead to a goal state.
Verification problem for while-free KBPs

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

**Proof idea.**

- **Upper bound:** on-the-fly model checking;
- **Lower bound:** reduction from TQBF.

<table>
<thead>
<tr>
<th>Agent</th>
<th>Value of $p_1$</th>
<th>Value of $p_2$</th>
<th>Value of $p_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Verification problem for while-free KBPs

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.

agent 1

\[ \text{value of } p_1 \]

agent 2

\[ \text{value of } p_2 \]

agent 3

\[ \text{value of } p_3 \]
The verification problem for while-free KBPs is PSPACE-complete.

**Proof idea.**

- **Upper bound**: on-the-fly model checking;
- **Lower bound**: reduction from TQBF.
Verification problem for general KBPs

**Theorem**

*The verification problem for general KBPs is undecidable.*

**Proof Idea.** Reduction from the halting problem of a Turing machine on input ε.
Outline

1. Knowledge-based programs
2. Semantics
3. Mathematical properties
   - Verification
   - Execution problem
   - Succinctness
4. Conclusion
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.
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 $a$;
- an action $act$.

Output: yes, if the next action of agent $a$ is $act$; no otherwise.
Reactive policy representation

Definition (reactive policy representation)

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

Example (Tree policies are reactive policy representation)

```
if justobserved(🔥) then turn left else stay
```

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

**Proposition**

*The execution problem for KBPs is PSPACE-complete.*
Outline

1. Knowledge-based programs
2. Semantics
3. Mathematical properties
   - Verification
   - Execution problem
   - Succinctness
4. Conclusion
Modal depth

Modal depth = number of nested ‘... knows ’ operators.

<table>
<thead>
<tr>
<th>Formulas</th>
<th>Modal depths</th>
</tr>
</thead>
<tbody>
<tr>
<td>justobserved(🔥)</td>
<td>0</td>
</tr>
<tr>
<td>a knows p</td>
<td>1</td>
</tr>
<tr>
<td>a knows (b knows p)</td>
<td>2</td>
</tr>
</tbody>
</table>
Succinctness

Theorem ([Lang, Zanuttini, 2012] for \( d = 1 \); [AAAI2018], for \( d > 1 \))

Let \( d \geq 1 \).

There is a poly\((n)\)-size QdecPOMDP family \((\mathcal{M}_{n,d})_{n\in\mathbb{N}}\) 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 \( \text{NP} \not\subseteq \text{P/poly} \), for any reactive policy representations, no poly\((n)\)-size valid policy family.
Succinctness

**Theorem (Lang, Zanuttini, 2012) for \( d = 1 \); [AAAI2018], for \( d > 1 \)**

Let \( d \geq 1 \).

There is a \( \text{poly}(n) \)-size QdecPOMDP family \( (\mathcal{M}_{n,d})_{n \in \mathbb{N}} \) for which:

1. there is a \( d \)-modal depth \( \text{poly}(n) \)-size valid KBP family;
2. no \( (d - 1) \)-modal depth valid KBP family;
3. assuming \( \text{NP} \not\subseteq \text{P/poly} \), for any reactive policy representations, no \( \text{poly}(n) \)-size valid policy family.

**Proof idea.** \( \mathcal{M}_{n,d} \):

- run a \( \text{poly}(n) \)-time protocol revealing a \( \text{poly}(n) \)-size 3-CNφ \( \beta \);
- \( \beta \) satisfiable iff a \( d \)-md non \( d - 1 \)-md expressible epistemic property holds.
Outline

1. Knowledge-based programs
2. Semantics
3. Mathematical properties
4. Conclusion
Conclusion

Model → Planning → Goal

- a’s KBP → a’s reactive policy
- b’s KBP → b’s reactive policy
- c’s KBP → c’s reactive policy

Knowledge-based programs
Semantics
Mathematical properties
Conclusion
Perspectives

- Implementation of the verification problem;
- Heuristics for the planning problem;
- More tractable fragments;
- decPOMDP (with probabilities);
- Temporal properties;
- Strategic reasoning;
- Develop proof systems for KBPs. Use of Coq?
Coming soon... New graphics for Hintikka’s world...

Feel free to use it!
http://people.irisa.fr/Francois.Schwarzentruber/hintikkasworld/