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

Model

Goal

Planning

a’s program

b’s program

c’s program
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;
D_Work DWork;

void fixed_step(void)
{
    int16_T FilterCoefficient_m;
    FilterCoefficient_m = ((int16_T)((int32_T)(int16_T)(8403L * (int32_T)error >> 13U) - DWork.Filter_DSTATE) << 4U) * 17999L >> 14);
    torque_request = (((int16_T)(1247L * (int32_T)error >> 14U) >> 1) +
        (DWork.Integrator_DSTATE >> 2)) + (FilterCoefficient_m >> 1);
    DWork.Integrator_DSTATE = ((int16_T)((int32_T)(1443L * (int32_T)error >> 13U) + 5243L >> 19U) + DWork.Integrator_DSTATE);
    DWork.Filter_DSTATE = ((int16_T)(5243L * (int32_T)FilterCoefficient_m >> 16U) +
        DWork.Filter_DSTATE);
}

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

- **KBP for agent a**
  - listenRadio
  - if \textit{a knows strike} then toStation
  - else toAirport

- **KBP for agent b**
  - readNewsPaper
  - if \textit{b knows strike} then toStation
  - else toAirport

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

Extends: \cite{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

Example (knowledge-based program for agent $a$)

```java
if $a$ knows (door 12 is locked and justobserved(🔥)) then
    turn left
    broadcast temperature
else
    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:

\[ \begin{align*}
  a: & \quad \text{stay} \\ b: & \quad \text{turn left}
\end{align*} \quad \xrightarrow{\text{state1}} \quad \text{state2} \]

\[ \begin{align*}
  a: & \quad \text{fire} \\ b: & \quad \text{cloud}
\end{align*} \quad \xrightarrow{\text{state1}} \quad \text{state2}
\]

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

Worlds = consistent histories of the form

$\overrightarrow{s_0pc_0obs^1s^1pc^1} \ldots \overrightarrow{obs^Ts^Tpc^T}$

where

$\overrightarrow{obs^t}$ vector of observations at time $t$

$\overrightarrow{s^t}$ state at time $t$

$\overrightarrow{pc^t}$ vector of program counters at time $t$
Epistemic structures at time $t$: indistinguishability relations

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

$$s^0 \xRightarrow{p^c_0} obs^1 s^1 \xRightarrow{p^c_1} \ldots \xRightarrow{obs^T s^T p^c T}$$

$$s'^0 \xRightarrow{p^c'_0} obs'^1 s'^1 \xRightarrow{p^c'_1} \ldots \xRightarrow{obs'^T s'^T p^c'_T}$$

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

$$\xrightarrow{obs^t_a} = \xrightarrow{obs'^t_a}$$
Program counters

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

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

- $(T, \text{start}, \bullet)$
- $(T, \text{listenRadio}, \blacksquare)$
- $(K_a \text{strike}, \text{toStation}, \blacktriangle)$
- $(\neg K_a \text{strike}, \text{toAirport}, \blacktriangle)$
Control-flow graph

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

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

$\left( T, \text{listenRadio}, \blacksquare \right)$

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

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

In the QdecPOMDP:

\[
\begin{align*}
S^0 \xrightarrow{\text{listenRadio}} S^1 \\
S^1 \xrightarrow{\text{toStation}} S^2
\end{align*}
\]

KBP control-flow graph

\[
\begin{align*}
(T, \text{start}, \bullet) \\
&\quad \downarrow \\
&\quad \uparrow \\
(T, \text{listenRadio}, \Box) \\
&\quad \downarrow \\
(K_a \text{strike}, \text{toStation}, \triangledown) & (\neg K_a \text{strike}, \text{toAirport}, \triangledown)
\end{align*}
\]
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.

\[
\begin{align*}
\text{agent 1} & \quad \text{value of } p_1 \\
\text{agent 2} & \quad \text{value of } p_2 \\
\text{agent 3} & \quad \text{value of } p_3
\end{align*}
\]
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: value of $p_1$

agent 2: value of $p_2$

agent 3: value of $p_3$
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.

<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>Agent 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agent 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agent 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.

\[
\begin{align*}
\text{agent 1} & \quad \text{value of } p_1 \\
\text{agent 2} & \quad \text{value of } p_2 \\
\text{agent 3} & \quad \text{value of } p_3
\end{align*}
\]
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 $\epsilon$. 

[Diagram showing a Turing machine with an input $\epsilon$]
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)

```plaintext
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-CNF $\beta$;
- $\beta$ satisfiable iff a $d$-md non $d - 1$-md expressible epistemic property holds.
Conclusion

Knowledge-based programs
Semantics
Mathematical properties
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
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/