# `Aximo`*

## automated axiomatic reasoning for information update

S. Richards & M. Sadrzadeh**

ECS, Southampton & PPS, Paris 7

* http://www.charcoalfeathers.net/research/projects/aximo/

** http://www.pps.jussieu.fr/∼mehrs/

what is `Aximo`?

a program that automatically verifies

*information update* properties

of <u>interactive multi-agent systems</u>:

- agents have some initial knowledge,
- they communicate,
- as a result acquire new knowledge.

what is the syntax of these properties?

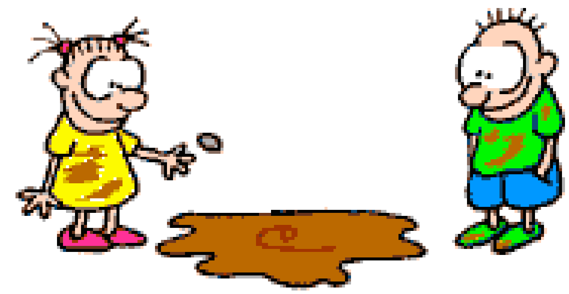
| initial knowledge | communication action |
|---|---|
| $\Box_A m$ | $a$ |

knowledge after communication

$$[a]\Box_A m$$

Example: coin toss

| Initially | After `honest` action | After `dishonest` action |
|---|---|---|
| $\Box_A(H \vee T)$ | $[H!]\Box_A H$ | $[H\dagger]\Box_A H$ |
| $\Box_A\Box_B(H \vee T)$ | $[H!]\Box_A\Box_B H$ | $[H\dagger]\Box_A\Box_B H$ |
| $\Box_A\Box_C(H \vee T)$ | $[H!]\Box_A\Box_C H$ | $[H\dagger]\Box_A\Box_C H$ |
| no knowledge | correct knowledge | wrong knowledge |



The muddy children puzzle

After father's

$$[a_0]\Box_j \bigvee_{i=1}^{k} D_i$$

After $k-1$ rounds

$$[a_0 \cdot \underbrace{a \cdot \cdots \cdot a}_{k-1}]\Box_i D_i$$

After $k-1$ rounds and `lying` of dirty children in round $k$

$$[a_0 \cdot \underbrace{a \cdot \cdots \cdot a}_{k-1} \cdot \overline{a}]\Box_j D_j$$

What kind of logic expresses these kind of properties?

a logic with two levels:

propositions

$$m ::= m \wedge m \mid m \vee m \mid \Box_A m \mid [a]m$$

actions

$$a ::= a \vee a' \mid a \cdot a' \mid 1$$

$$\text{Dynamic Epistemic logic} \begin{cases} \text{PDL with epistemic modalities} \\ \text{Epistemic logic with actions} \end{cases}$$

Semantics for epistemics:

Kripke semantics

$$(S, R_A \subseteq S \times S)_{\Phi}$$

$$s \models \Box_A \phi \quad \text{iff} \quad \forall t, (s,t) \in R_A, \quad t \models \phi$$

Algebraic semantics

$$(\mathcal{P}(S), app_A \colon \mathcal{P}(S) \to \mathcal{P}(S))$$

$$m \subseteq \Box_A m' \quad \text{iff} \quad app_A(m) \subseteq m'$$

$$m \leq \Box_A m' \quad \text{iff} \quad app_A(m) \leq m'$$

Semantics for dynamics:

Kripke semantics

$$(S, R_A \subseteq S \times S)_{\Phi} \quad \cdot \quad (\Sigma, R_A \subseteq \Sigma \times \Sigma)_{\Phi'}$$

$$s \models \Box_A \phi \quad \text{iff} \quad \forall t, (s,t) \in R_A, \quad t \models \phi$$

$$s \models [\sigma]\Box_A \phi \quad \text{iff} \quad (s, \sigma) \models \Box_A \phi$$

Algebraic semantics

$$(\mathcal{P}(S^*), app_A \colon \mathcal{P}(S^*) \to \mathcal{P}(S^*)) \quad \cdot \quad (\mathcal{P}(\Sigma^*), app_A \colon \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^*))$$

$$m \leq \Box_A m' \quad \text{iff} \quad app_A(m) \leq m'$$

$$m \leq [a]\Box_A m' \quad \text{iff} \quad m \cdot a \leq \Box_A m'$$

Interpretation:

$app_A(m)$: appearance of $m$ to agent $A$

All the propositions that appear to agent $A$ as true when $m$ holds.

$\Box_A m$ agent $A$ knows that $m$.

$app_A(\sigma)$: appearance of $\sigma$ to $A$

All the actions that appear to agent $A$ as happening when $m$ is happening.

$\Box_A a$: agent $A$ knows that action $a$ is happening.

$[a]\Box_A m$: after action $a$ agent $A$ knows that $m$.

# Algebraic Axiomatics

Definition. An **Epistemic System** $\mathcal{E} = (M, Q, \{app_A\}_{A\in\mathcal{A}})$ is

$$\begin{cases} \text{a quantale} & (Q, \bullet, \epsilon, \bigvee) \\ \text{its right module} & (M, \bigvee, \cdot) \\ \text{a family of endos} & app_A = (app_A^M : M \to M, app_A^Q : Q \to Q) \end{cases}$$

satisfying:

$$\begin{aligned} \epsilon &\leq app_A^Q(\epsilon) \\ app_A^Q(a \bullet a') &\leq app_A^Q(a) \bullet app_A^Q(a') \\ app_A^M(m \cdot a) &\leq app_A^M(m) \cdot app_A^Q(a) \\ Stab(Q) &= \{m \in M \mid \forall a \in Q, m \cdot a \leq m\} \\ \forall a \in Q, ker(a) &= \{m \in M \mid m \cdot a = \bot\} \end{aligned}$$

# Example


$$(M, Q, \{app_A\}_{A \in \mathcal{A}})$$

$A, B, C \in \mathcal{A}$
$s, t \in M$
$H, T \in Stab(Q)$
$H!, H\dagger \in Q$
$t \leq ker(H!), s \leq ker(H\dagger)$

$$s \leq [H!]\Box_A H, \qquad t \leq [H\dagger]\Box_C T, \qquad t \leq [H\dagger]\Box_A H$$

# Muddy Children

$$(M, Q, \{app_A\}_{A \in \mathcal{A}})$$

$\{C_1, \cdots, C_n\} \subseteq \mathcal{A}$ set of children, $k$ of them dirty,

$s_\beta \in M$ all possible initial states for $\beta$ a subset of children,

$app^M_{C_i}(s_\beta) = s_{\beta \setminus \{C_i\}} \vee s_{\beta \cup \{C_i\}}$,

$\{D_\emptyset\} \cup \{D_i \mid C_i \in \mathcal{A}\} \subseteq Stab(Q)$,

$s_\beta \leq D_i$ for $C_i \in \beta$, $S_\emptyset \leq D_\emptyset$,

$q_0 \in Q$ with kernel $D_\emptyset$,

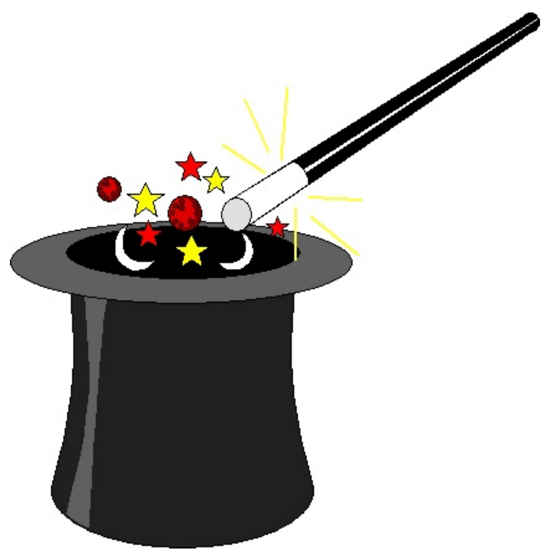
$q \in Q$ with kernel $\bigvee_{i=1}^{i=n} \Box_{C_i} D_i$.

# Muddy Children

**Proposition.**
After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0\,(\cdot q)^{(k-1)}]\Box_{C_j} D_j.$$

**Proof.**



## Muddy Children

**Proposition.**
After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0\,(\cdot q)^{(k-1)}]\Box_{C_j} D_j.$$

## Muddy Children

**Proposition.**

After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0\,(\cdot q)^{(k-1)}]\Box_{C_j} D_j.$$

$$s_{\{C_1,\cdots,C_k\}} \cdot (q_0\,(\cdot q)^{(k-1)}) \leq \Box_{C_j} D_j$$

## Muddy Children

**Proposition.**
After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0\,(\cdot q)^{(k-1)}]\Box_{C_j} D_j.$$

$$s_{\{C_1,\cdots,C_k\}} \cdot (q_0\,(\cdot q)^{(k-1)}) \leq \Box_{C_j} D_j$$

$$(s_{\{C_1,\cdots,C_k\}} \cdot q_0)\,(\cdot q)^{(k-1)} \leq \Box_{C_j} D_j$$

# Muddy Children

**Proposition.**

After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0\,(\cdot q)^{(k-1)}]\Box_{C_j} D_j.$$

$$s_{\{C_1,\cdots,C_k\}} \cdot (q_0\,(\cdot q)^{(k-1)}) \leq \Box_{C_j} D_j$$

$$(s_{\{C_1,\cdots,C_k\}} \cdot q_0)\,(\cdot q)^{(k-1)} \leq \Box_{C_j} D_j$$

# Muddy Children

**Proposition.**
After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0 \, (\cdot q)^{(k-1)}]\Box_{C_j} D_j.$$

$$s_{\{C_1,\cdots,C_k\}} \cdot \left(q_0 \, (\cdot q)^{(k-1)}\right) \leq \Box_{C_j} D_j$$

$$\left(s_{\{C_1,\cdots,C_k\}} \cdot q_0\right) (\cdot q)^{(k-1)} \leq \Box_{C_j} D_j$$

$$app_{C_j}\left(\left(s_{\{C_1,\cdots,C_k\}} \cdot q_0\right) (\cdot q)^{(k-1)}\right) \leq D_j$$

# Muddy Children

**Proposition.**
After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0\,(\cdot q)^{(k-1)}]\Box_{C_j} D_j.$$

$$s_{\{C_1,\cdots,C_k\}} \cdot \left(q_0\,(\cdot q)^{(k-1)}\right) \leq \Box_{C_j} D_j$$

$$\left(s_{\{C_1,\cdots,C_k\}} \cdot q_0\right)(\cdot q)^{(k-1)} \leq \Box_{C_j} D_j$$

$$app_{C_j}\left(\left(s_{\{C_1,\cdots,C_k\}} \cdot q_0\right)(\cdot q)^{(k-1)}\right) \leq D_j$$

$$app_{C_j}(s_{\{C_1,\cdots,C_k\}}) \cdot q_0\,(\cdot q)^{(k-1)} \leq D_j$$

# Muddy Children

**Proposition.**
After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0\,(\cdot q)^{(k-1)}]\Box_{C_j} D_j.$$

$$s_{\{C_1,\cdots,C_k\}} \cdot (q_0\,(\cdot q)^{(k-1)}) \leq \Box_{C_j} D_j$$

$$(s_{\{C_1,\cdots,C_k\}} \cdot q_0)\,(\cdot q)^{(k-1)} \leq \Box_{C_j} D_j$$

$$app_{C_j}\left((s_{\{C_1,\cdots,C_k\}} \cdot q_0)\,(\cdot q)^{(k-1)}\right) \leq D_j$$

$$app_{C_j}(s_{\{C_1,\cdots,C_k\}}) \cdot q_0\,(\cdot q)^{(k-1)} \leq D_j$$

$$(s_{\{C_1,\cdots,C_k\}} \vee s_{\{C_1,\cdots,C_k\}\setminus\{C_j\}}) \cdot q_0\,(\cdot q)^{(k-1)} \leq D_j$$

# Muddy Children

**Proposition.**
After $k-1$ rounds of refutations, child $j$ knows that he is dirty

$$s_{\{C_1,\cdots,C_k\}} \leq [q_0 \, (\cdot q)^{(k-1)}] \Box_{C_j} D_j.$$

$$s_{\{C_1,\cdots,C_k\}} \cdot (q_0 \, (\cdot q)^{(k-1)}) \leq \Box_{C_j} D_j$$

$$(s_{\{C_1,\cdots,C_k\}} \cdot q_0) \, (\cdot q)^{(k-1)} \leq \Box_{C_j} D_j$$

$$app_{C_j} \left( (s_{\{C_1,\cdots,C_k\}} \cdot q_0) \, (\cdot q)^{(k-1)} \right) \leq D_j$$

$$app_{C_j}(s_{\{C_1,\cdots,C_k\}}) \cdot q_0 \, (\cdot q)^{(k-1)} \leq D_j$$

$$(s_{\{C_1,\cdots,C_k\}} \vee s_{\{C_1,\cdots,C_k\} \setminus \{C_j\}}) \cdot q_0 \, (\cdot q)^{(k-1)} \leq D_j$$

$$s_{\{C_1,\cdots,C_k\}} \cdot q_0 \, (\cdot q)^{(k-1)} \leq D_j$$

$$s_{\{C_1,\cdots,C_k\} \setminus \{C_j\}} \cdot q_0 \, (\cdot q)^{(k-1)} \leq D_j$$

## Lying children

assume $k$ dirty, after round $k-1$ of no answers, dirty children lie in round $k$ by still saying no. clean children know they are dirty.

# Aximo

... is based on the axioms of algebraic semantics

input: assumptions and property in form of inequalities

algorithm: re-write rules base on the in-equations of algebra

output: proof + yes-no answer

Syntax and Proof Rules

Decision Procedure and Normal Forms

Complexity and Termination

Semantics

Soundness and a-completeness

# Syntax

(I) Input inequality

$$l \leq r$$

generated by expressions

$$\begin{aligned} l &::= s \mid l.a \mid l_1 * l_2 \\ r &::= f \mid \triangle\, r \mid r_1 * r_2 \\ a &::= a_1 \vee a_2 \\ * &::= \wedge \mid \vee \\ \triangle &::= \Box_A \mid [a] \end{aligned}$$

(II) Kernel of actions

$$k ::= f \mid \Box_A f \mid k * k$$

(III) Appearance of states and actions

$$app_A(s) = \{s_1, \cdots, s_n\}, app_A(a) = \{a_1, \cdots, a_n\}$$

(IV) States satisfying facts

$$s \leq f$$

# Re-write rules

(I) For inequalities

$$l_1 \wedge l_2 \leq r_1 \vee r_2 \quad \rightsquigarrow \quad \begin{cases} l_1 \leq r_1 & \text{or} \\ l_1 \leq r_2 & \text{or} \\ l_2 \leq r_1 & \text{or} \\ l_2 \leq r_2 & . \end{cases} \tag{1}$$

$$l_1 \vee l_2 \leq r_1 \wedge r_2 \quad \rightsquigarrow \quad \begin{cases} l_1 \leq r_1 & \text{and} \\ l_1 \leq r_2 & \text{and} \\ l_2 \leq r_1 & \text{and} \\ l_2 \leq r_2 & . \end{cases} \tag{2}$$

$$l \leq \triangle\, r \quad \rightsquigarrow \quad \begin{cases} l \cdot a \leq r & \triangle = [a] \\ app_A(l) \leq r & \triangle = \Box_A \end{cases} \tag{3}$$

(II) For expressions

$$\triangle(r_1 * r_2) \quad \rightsquigarrow \quad (\triangle r_1) * (\triangle r_2) \tag{4}$$

$$(l_1 * l_2) \cdot a_1 \cdots a_n \quad \rightsquigarrow \quad (l_1 \cdot a_1 \cdots a_n) * (l_2 \cdot a_1 \cdots a_n) \tag{5}$$

$$l \cdot (a_1 \vee a_2) \quad \rightsquigarrow \quad (l \cdot a_1) \vee (l \cdot a_2) \tag{6}$$

$$[a_1 \vee a_2]\, r \quad \rightsquigarrow \quad [a_1]\, r \vee [a_2]\, r \tag{7}$$

$$app_A(l_1 * l_2) \quad \rightsquigarrow \quad app_A(l_1) * app_A(l_2) \tag{8}$$

$$app_A(l \cdot a_1 \cdots a_n) \quad \rightsquigarrow \quad app_A(l) \cdot app_A(a_1) \cdots app_A(a_n) \tag{9}$$

(III) Read from the input

$$app_A(s) \quad \rightsquigarrow \quad \vee_i^k s_i$$

$$app_A(a) \quad \rightsquigarrow \quad \vee_i^{k'} a_i$$

# A Decision Procedure

(I) Eliminate connectives

1. repeat until all $*$'s are eliminated, write to 'and' - 'or' lists
   (a) eliminate the $*$'s outside scope of a $\triangle$ or $- \cdot a$ by 1,2
   (b) push the $*$'s inside scope of a $\triangle$ or $- \cdot a$ to outside by 4-7
2. eliminate the $\triangle$'s by 3
3. push the $*$'s inside scope of $app_A$ to outside by repeating 8,9
4. eliminate $app_A$ by repeating: read input; do 2; write 'or'

| Normal Form: $s' \cdot a'_1 \cdots a'_n \leq f$ |
|---|

(II) Eliminate inequalities

- For the first inequality in the 'and' list do
  - For each inequality in the 'or' list do
    1. if $s' \leq f$ then eliminate
    2. else repeat $\begin{cases} \text{recursively call} & s' \leq ker(a'_1) \\ \text{recursively call} & s' \cdot a'_1 \leq ker(a'_2) \\ \cdots & \\ \text{recursively call} & s' \cdot a'_1 \cdots a'_{n-1} \leq ker(a'_n) \end{cases}$
  - If 'or' list empty return 'expression passed', stop.
- else move to the next inequality in the 'and' list, repeat.
- return 'failed to prove'.

# Examples

$$s(0) \leq e(0)(f(0) \wedge f(1)) \vee f(2)$$


Honest

$$\begin{aligned} s(0) &\leq e(1)f(0) \\ s(0) &\leq d(0)e(1)f(0) \\ s(0) &\leq d(0)e(0)f(0) \end{aligned}$$

Lying

$$\begin{aligned} s(0) &\leq d(3)e(1)f(0) \\ s(0) &\leq d(3)e(0)f(0) \end{aligned}$$

# Complexity

$$W \times W' \times W''$$

$$W = (\sharp\wedge+1)\times(\sharp\vee+1), W' = (k^{\sharp\Box})^{1+\sharp[-]}, W'' = \Sigma_{i=0}^{\sharp[-]} (k^{\sharp\Box\in Ker})^i$$

for

$$k = max\,\{|\,app_A(s)\,|, |\,app_A(a_1)\,|, \cdots, |\,app_A(a_n)\,|\}$$

Simplifies to the order of

$$(k^{2\times max\sharp\Box})^{\sharp[-]+1}$$

If no one cheats or no one suspects the cheating

$$k^{2\times max\sharp\Box} \times (\sharp[-])$$

Proposition. **Termination and Decidability**
There are $W \times W' \times W''$ inequalities to be verified and all the recursions are terminating since kernels do not have dynamic modalities.

Expressions of `Aximo` are interpreted in an epistemic system

$$[\![-]\!]^M : \mathcal{L}_{l,r} \to M, \qquad [\![-]\!]^Q : \mathcal{L}_a \to Q$$

by induction, e.g.

$$[\![s \cdot a]\!]^M = [\![s]\!]^M \cdot [\![a]\!]^Q, \quad [\![l_1 * l_2]\!]^M = [\![l_1]\!]^M * [\![l_2]\!]^M$$

$$[\![\Box_A r]\!]^M = \Box_A [\![r]\!]^M, \quad [\![[a]r]\!]^M = [[\![a]\!]^Q][\![r]\!]^M$$

where

$$app_A \dashv \Box_A, \qquad - \cdot a \dashv [a]-$$

Theorem. **Soundness**

For a set of assumption inequalities $\mathcal{S}$, if an inequality holds in `Aximo`, then it holds in all epistemic systems $\mathcal{E}$ in which $\mathcal{S}$ holds.

Theorem. **a-Completeness**

For a set of assumption inequalities $\mathcal{S}$, if an `Aximo`-expressible inequality does not hold in an $\mathcal{E}$ in which $\mathcal{S}$ holds, then it does not hold in `Aximo`.

**Existence of models**

One way to build an $\mathcal{E}$ in which $\mathcal{S}$ holds is to first form a Kripke model from the inequalities in $\mathcal{S}$ and then apply the construction mentioned in previous work.

# Honest muddy children

Child 0, 1 and 2 are dirty:

After initial announcement and two rounds of null answers, child 0 knows he is dirty:

$$s(7) \leq d(0)d(1)d(1)e(0)f(5)$$

Lying muddy children

Child 0, 1, 2 are dirty, but give a null answer in round 3 :
After initial announcement, two rounds of real null answers and the extra lying null answer,
child 3 'knows' he is dirty (even though he is actually clean):

$$s(7) \leq d(0)d(1)d(1)d(1)e(3)f(8)$$

# Future work

1- Make all rules invertible:

$$l \leq \Box_A(r_1 \vee r_2) \quad \leadsto \quad l \leq \Box_A r_1 \vee \Box_A r_2$$

to

$$l \leq \Box_A(r_1 \vee r_2) \quad \leadsto \quad app_A(l) \leq r_1 \vee r_2$$

How about the lattice rules? ... using dist

2- What kind of algebra the terms of `Aximo` form?

or,

wrt to what algebraic semantics `Aximo` is complete?