Knowledge in the Situation Calculus

Adrian Pearce

8 July 2009

includes slides by Ryan Kelly

Outline

Outline

Knowledge

Extensions to the Situation Calculus for representing and reasoning about knowledge

- Reasoning about knowledge with hidden actions
- Reasoning about group-level knowledge modalities

Explanation closure assumes complete knowledge of $\mathcal{D}_{ssa}$

- Golog assumes complete knowledge of $\mathcal{D}_{ad}$ and $\mathcal{D}_{una}$ in S_0
- What if incomplete knowledge: $\quad \mathbf{Knows}(\phi, s)$?

Basic Action Theory (Revisited)

Definition (Basic Action Theory)

A basic action theory, denoted $\mathcal{D}$, consists of:

- the foundational axioms of the situation calculus (Σ);
- action description axioms such as preconditions ($\mathcal{D}_{ad}$);
- successor state axioms describing how primitive fluents change between situations ($\mathcal{D}_{ssa}$);
- axioms describing the initial situation ($\mathcal{D}_{S_0}$);
- and axioms describing background facts ($\mathcal{D}_{bg}$)

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ad} \cup \mathcal{D}_{ssa} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{bg}$$

Basic Action Theory (Revisited)

Definition (Basic Action Theory)

A basic action theory, denoted $\mathcal{D}$, consists of:

- the foundational axioms of the situation calculus (Σ);
- action description axioms such as preconditions ($\mathcal{D}_{ad}$);
- successor state axioms describing how primitive fluents change between situations ($\mathcal{D}_{ssa}$);
- axioms describing the initial situation ($\mathcal{D}_{S_0}$);
- and axioms describing background facts ($\mathcal{D}_{bg}$)

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ad} \cup \mathcal{D}_{ssa} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{bg}$$

- Regression operator performs induction over Σ, $\mathcal{D}_{ssa}$ and $\mathcal{D}_{bg}$ resulting in query $\mathcal{D}_{bg} \cup \mathcal{D}_{S_0}$

Basic Action Theory (Revisited)

Definition (Basic Action Theory)

A basic action theory, denoted $\mathcal{D}$, consists of:

- the foundational axioms of the situation calculus (Σ);
- action description axioms such as preconditions ($\mathcal{D}_{ad}$);
- successor state axioms describing how primitive fluents change between situations ($\mathcal{D}_{ssa}$);
- axioms describing the initial situation ($\mathcal{D}_{S_0}$);
- and axioms describing background facts ($\mathcal{D}_{bg}$)

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ad} \cup \mathcal{D}_{ssa} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{bg}$$

- Regression operator performs induction over Σ, $\mathcal{D}_{ssa}$ and $\mathcal{D}_{bg}$ resulting in query $\mathcal{D}_{bg} \cup \mathcal{D}_{S_0}$
- Complete knowledge of $\mathcal{D}_{ad}$, $\mathcal{D}_{bg}$ and $\mathcal{D}_{ssa}$ assumed.

Outline

Limitation: Synchronicity

This works well, but it depends on two assumptions:

- Complete knowledge (linear plan, no sensing)
- Synchronous domain (agents proceed in lock-step)

Nearly universal in the literature: "assume all actions are public".

Challenge: Regression depends intimately on synchronicity

Two aspects to knowledge

Two aspects to knowledge

- incomplete information (through action can learn)
- lack of synchronisation (don't know how many actions have occurred)

Example: Alternating Bit Protocol

Protocol for S:

i:=0

while true do

begin read x_i;

send x_i

i := i+1

end

Protocol for R:

when $K_R(x_0)$ set i:=0

while true do

begin write x_i;

i:= i+1

end

Example: Alternating Bit Protocol

Protocol for S:

i:=0

while true do

begin read x_i;

send x_i until $K_S K_R(x_i)$;

i := i+1

end

Protocol for R:

when $K_R(x_0)$ set i:=0

while true do

begin write x_i;

send "$K_R(x_i)$"

i:= i+1

end

Example: Alternating Bit Protocol

Protocol for S:

i:=0
while true do
begin read x_i;
send x_i until $K_S K_R(x_i)$;
send "$K_S K_R(x_i)$" until $K_S K_R K_S K_R(x_i)$
i := i+1
end

Protocol for R:

when $K_R(x_0)$ set i:=0
while true do
begin write x_i;
send "$K_R(x_i)$" until $K_R K_S K_R(x_i)$;
send "$K_R K_S K_R(x_i)$" until $K_R(x_{x+1})$
i:= i+1
end

Equivalence relations

Definition (Kripke Models)

A Kripke model M is a tuple $\langle S, V, R_1, \ldots, R_m \rangle$ where:

Equivalence relations

Definition (Kripke Models)

A Kripke model M is a tuple $\langle S, V, R_1, \ldots, R_m \rangle$ where:

1. S is a non-empty set of *states*, *possible worlds* or *epistemic alternatives*,

Equivalence relations

Definition (Kripke Models)

A Kripke model M is a tuple $\langle S, V, R_1, \ldots, R_m \rangle$ where:

1. S is a non-empty set of *states*, *possible worlds* or *epistemic alternatives*,
2. $V : S \to (p \to \{true, false\})$ is a truth assignment to the propositional atoms (p) per state,

Equivalence relations

Definition (Kripke Models)

A Kripke model M is a tuple $\langle S, V, R_1, \ldots, R_m \rangle$ where:

1. S is a non-empty set of *states*, *possible worlds* or *epistemic alternatives*,
2. $V : S \to (p \to \{true, false\})$ is a truth assignment to the propositional atoms (p) per state,
3. $R_i \subseteq S \times S$ (for all $i \in A$) are the *epistemic accessibility* relations for each agent.

Equivalence relations

Definition (Kripke Models)

A Kripke model M is a tuple $\langle S, V, R_1, \ldots, R_m \rangle$ where:

1. S is a non-empty set of *states*, *possible worlds* or *epistemic alternatives*,
2. $V : S \rightarrow (p \rightarrow \{true, false\})$ is a truth assignment to the propositional atoms (p) per state,
3. $R_i \subseteq S \times S$ (for all $i \in A$) are the *epistemic accessibility* relations for each agent.

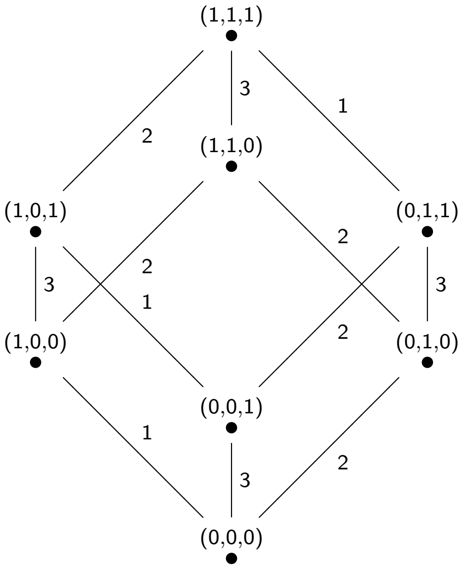
For any state or possible world s,
$(M, s) \models p$ (for $p \in P$) iff $V(s)(p) = \text{true}$

Example: Muddy Children Puzzle

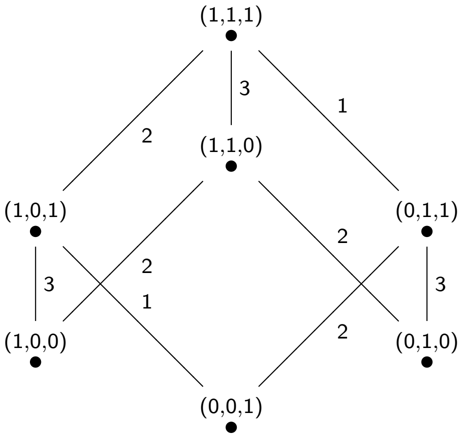
Example

- k children get mud on their foreheads
- Each can see the mud on others, but not on his/her own forehead
- The father says *at least one of you had mud on your head"* initially.
- The father then repeats *Can any of you prove you have mud on your head?* over and over.
- Assuming that the children are perceptive, intelligent, truthful, and that they answer simultaneously, what will happen?

Muddy Children Puzzle (Initially)



Muddy Children Puzzle (After the father speaks)



Forms of knowledge

- $D_G p$: the group G has distributed knowledge of fact p
- $S_G p$: someone in G knows p

$$S_G p \equiv \vee_{i \in G} K_i p$$

- $E_G p$: everyone in G knows p

$$E_G p \equiv \wedge_{i \in G} K_i p$$

Forms of knowledge

- $E_G^k p$ for $k \geq 1$: $E_G^k p$ is defined by

$$E_G^1 p = E_G p$$

$$E_G^{k+1} p = E_G E_G^k p \text{ for } k \geq 1$$

- $C_G p$: p is common knowledge in G

$$C_G \equiv E_G p \wedge E_G^2 p \wedge \ldots E_G^m p \wedge \ldots$$

Synchronisation (Revisited)

Example (The Coordinated Attack Problem (Byzantine Generals))

- Suppose General A sends a message to General B saying *Let's attack at Dawn*.
- Does not have any common knowledge fixpoint (in spite of acknowledgements).
- It seems that common knowledge is theoretically unachievable - how can this be so?

- In the presence of unreliable communication, common knowledge is theoretically unachievable.

Simultaneity

In practice, we can establish ϵ-common knowledge, Halpern and Moses (1990).

Definition (ϵ-common knowledge)

ϵ-common knowledge assumes that within an interval ϵ everybody knows ϕ.

Knowledge

Agent i knows p in world s of (Kripke) structure M, exactly if p is true at all worlds that i considers possible in s. Formally,

Knowledge

Agent i knows p in world s of (Kripke) structure M, exactly if p is true at all worlds that i considers possible in s. Formally,

$(M, s) \models K_i p$ iff $(M, t) \models p$ for all t such that $(s, t) \in K_i$

Knowledge

Agent i knows p in world s of (Kripke) structure M, exactly if p is true at all worlds that i considers possible in s. Formally,

$(M, s) \models K_i p$ iff $(M, t) \models p$ for all t such that $(s, t) \in K_i$

Relationship between knowledge forms, D_G, E_G and C_G:

Knowledge

Agent i knows p in world s of (Kripke) structure M, exactly if p is true at all worlds that i considers possible in s. Formally,

$(M, s) \models K_i p$ iff $(M, t) \models p$ for all t such that $(s, t) \in K_i$

Relationship between knowledge forms, D_G, E_G and C_G:

- $\models E_G p \Leftrightarrow \wedge_{i \in G} K_i p$

Knowledge

Agent i knows p in world s of (Kripke) structure M, exactly if p is true at all worlds that i considers possible in s. Formally,

$(M, s) \models K_i p$ iff $(M, t) \models p$ for all t such that $(s, t) \in K_i$

Relationship between knowledge forms, D_G, E_G and C_G:

- $\models E_G p \Leftrightarrow \wedge_{i \in G} K_i p$
- The notions of group knowledge form a hierarchy

$$C_G\varphi \supset \ldots \supset E_G^{k+1}\varphi \supset \ldots \supset E_G\varphi \supset S_G\varphi \supset D_G\varphi \supset \varphi$$

The properties of Knowledge (S5 axioms)

1. $K_i\varphi \wedge K_i(\varphi \Rightarrow \Psi)) \Rightarrow K_i \; \Psi$ (Distribution axiom)
2. *if* $M \models \varphi$ *then* $M \models K_i$ (Knowledge generalisation rule)
3. $K_i\varphi \Rightarrow \varphi$ (Knowledge or truth axiom)
4. $K_i\varphi \Rightarrow K_iK_i\varphi$ (Positive introspection axiom)
5. $\neg K_i\varphi \Rightarrow K_i\neg K_i\varphi$ (Negative introspection axiom)

View-based knowledge interpretations

- View-based knowledge interpretations, Halpern and Moses (1990): similar to Kripke structures in that have the properties of $S5$, additionally
- C1. The *fixed point axiom* $\models C_G p \Leftrightarrow E_G(p \wedge C_G p)$
- C2. the *induction rule* $p \supset E_G(p \wedge q)$ infer $p \supset C_G q$
- When views (of each agent) are indistinguishable (via equivalence relations) then common knowledge has been established: common knowledge can be induced, or C_G.

View-based knowledge interpretations

A *view-based knowledge interpretation* I is a triple (R, π, v), consisting of a set of runs R, an assignment π that associates with every point in R a truth assignment to the ground facts.
For every point $(r, t) \in R$ and every ground fact $p \in P$, we have

$$\pi(r, t)(P) \in \{true, false\}$$

and a view function v for R.

Outline

Observations

First, we must represent asynchronicity.
We reify the *observations* made by each agent, by adding the following action description function of the following form to D_{ad}:

$$Obs(agt, c, s) = o$$

If $Obs(agt, c, s) = \{\}$ then the actions are completely hidden.

$View(agt, S_0) = \epsilon$
$Obs(agt, c, s) = \{\} \rightarrow View(agt, do(c, s)) = View(agt, s)$
$Obs(agt, c, s) \neq \{\} \rightarrow View(agt, do(c, s)) = Obs(agt, c, s) \cdot View(agt, s)$

Observations

In synchronous domains, everyone observes every action:

$$a \in Obs(agt, c, s) \;\equiv\; a \in c$$

Sensing results can be easily included as action#sensing pairs:

$$a\#r \in Obs(agt, c, s) \;\equiv\; a \in c \wedge SR(a, s) = r$$

And observability can be axiomatised explicitly

$$a \in Obs(agt, c, s) \;\equiv\; a \in c \wedge CanObs(agt, a, s)$$

Observations

$$CanObs(agt, a, s) \equiv InSameRoom(agt, actor(a), s)$$

$$a \in Obs(agt, c, s) \equiv a \in c \wedge CanObs(agt, a, s) \\ \wedge \neg CanSense(agt, a, s)$$

$$a\#r \in Obs(agt, c, s) \equiv a \in c \wedge SR(a, s) = r \\ \wedge CanObs(agt, a, s) \wedge CanSense(agt, a, s)$$

$$CanSense(agt, activateSpeaker(agt_2), s) \equiv CloseToSpeaker(agt)$$

Observations

Action: global event changing the state of the world
Observation: local event changing an agent's knowledge

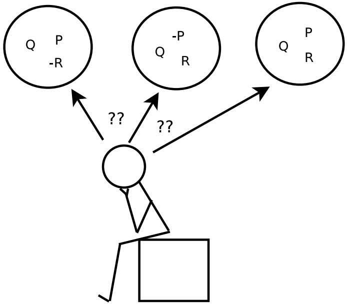
Situation: global history of actions giving current world state
View: local history of observations giving current knowledge

How can we let agents reason using only their local view?

Outline

Knowledge

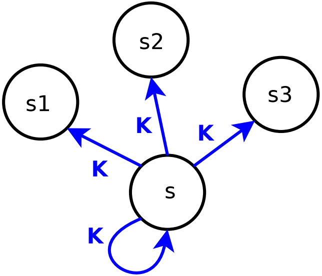
If an agent is unsure about the state of the world, there must be several different states of the world that it considers possible.
The agent *knows* ϕ iff ϕ is true in all possible worlds.



$$\textbf{Knows}(Q) \;\wedge\; \neg\textbf{Knows}(P) \;\wedge\; \neg\textbf{Knows}(R) \;\wedge\; \textbf{Knows}(P \vee R)$$

Knowledge

Introduce a possible-worlds fluent $K(agt, s', s)$:



We can then define knowledge as a simple macro:

$$\mathbf{Knows}(agt, \phi, s) \stackrel{\text{def}}{=} \forall s' \left[K(agt, s', s) \rightarrow \phi(s') \right]$$

Knowledge follows Observation

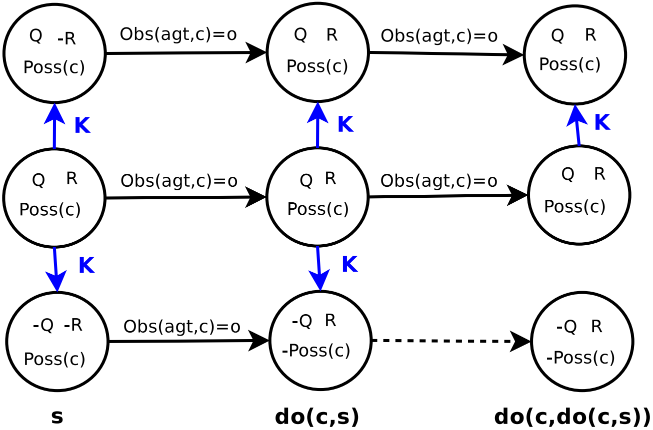
Halpern & Moses, 1990:
"an agent's knowledge at a given time must depend only on its local history: the information that it started out with combined with the events it has observed since then"

Clearly, we require:

$$K(agt, s', s) \equiv View(agt, s') = View(agt, s)$$

We must enforce this in the successor state axiom for K.

Knowledge: The Synchronous Case



Knowledge: The Synchronous Case

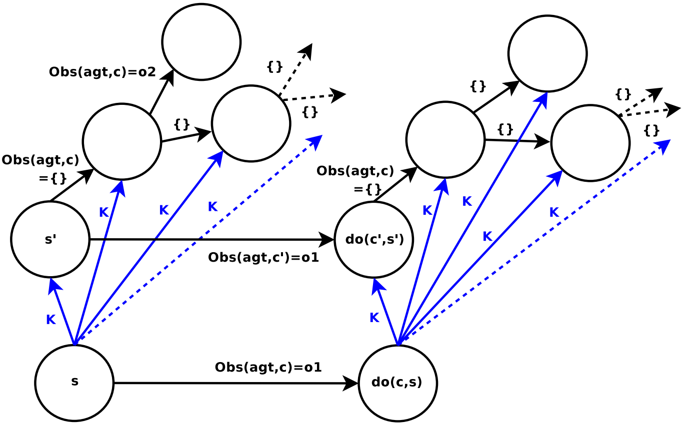
In the synchronous case, K_0 has a simple successor state axiom:

$$K_0(agt, s'', do(c, s)) \equiv \exists s', c' : \; s'' = do(c', s') \wedge K_0(agt, s', s) \\ \wedge Poss(c', s') \wedge Obs(agt, c, s) = Obs(agt, c', s')$$

And a correspondingly simple regression rule:

$$\mathcal{R}(\mathbf{Knows_0}(agt, \phi, do(c, s)) \stackrel{\text{def}}{=} \exists o : Obs(agt, c, s) = o \\ \wedge \forall c' : \mathbf{Knows_0}(agt, Poss(c') \wedge Obs(agt, c') = o \rightarrow \mathcal{R}(\phi, c'), s)$$

Knowledge: The Asynchronous Case



Knowledge: The Asynchronous Case

First, some notation:

$$s <_\alpha do(c, s') \ \equiv\ s \leq_\alpha s' \wedge \alpha(c, s')$$

$$PbU(agt, c, s) \stackrel{\text{def}}{=} Poss(c, s) \wedge Obs(agt, c, s) = \{\}$$

Then the intended dynamics of knowledge update are:

$$\begin{aligned} K(agt, s'', do(c, s)) \ \equiv\ & \exists o : Obs(agt, c, s) = o \\ & \wedge \left[o = \{\} \rightarrow K(agt, s'', s)\right] \\ & \wedge \left[o \neq \{\} \rightarrow \exists c', s' : K(agt, s', s)\right. \\ & \left.\wedge Obs(agt, c', s') = o \wedge Poss(c', s') \wedge do(c', s') \leq_{PbU(agt)} s''\right] \end{aligned}$$

Sync vs Async

We've gone from this:

$$K_0(agt, s'', do(c, s)) \equiv \exists s', c' : \; s'' = do(c', s') \land K_0(agt, s', s) \\ \land Poss(c', s') \land Obs(agt, c, s) = Obs(agt, c', s')$$

To this:

$$\begin{aligned} K(agt, s'', do(c, s)) \; \equiv \; \exists o : Obs(agt, c, s) = o \\ \land \left[o = \{\} \rightarrow K(agt, s'', s)\right] \\ \land \left[o \neq \{\} \rightarrow \exists c', s' : K(agt, s', s)\right. \\ \left. \land Obs(agt, c', s') = o \land Poss(c', s') \land do(c', s') \leq_{PbU(agt)} s''\right] \end{aligned}$$

It's messier, but it's also hiding a much bigger problem...

Regressing Knowledge

Our new SSA uses $\leq_{PbU(agt)}$ to quantify over all future situations. Regression cannot be applied to such an expression.

An asynchronous account of knowledge cannot be approached using the standard regression operator.

In fact, this quantification requires a second-order induction axiom. Must we abandon hope of an effective reasoning procedure?

Property Persistence (revisited)

Property persistence facilitates "factoring out" the quantification, this allows us to get on with the business of doing regression.

The *persistence condition* $\mathcal{P}[\phi, \alpha]$ of a formula ϕ and action conditions α to mean: assuming all future actions satisfy α, ϕ will remain true.

$$\mathcal{P}[\phi, \alpha](s) \;\equiv\; \forall s' : s \leq_\alpha s' \rightarrow \phi(s')$$

Like $\mathcal{R}$, the idea is to transform a query into a form that is easier to deal with.

Property Persistence

The persistence condition can be calculated as a fixpoint:

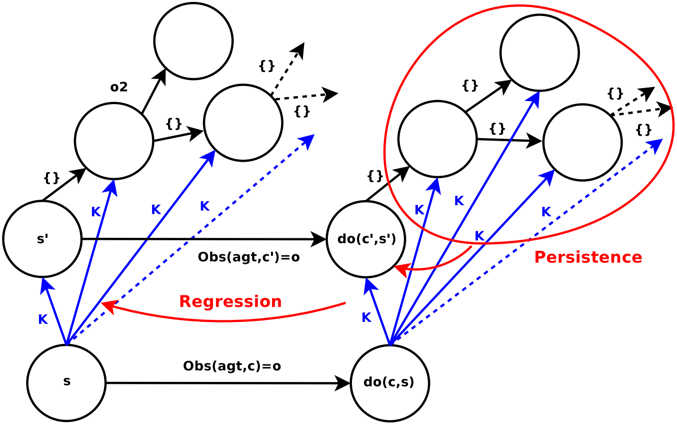
$$\mathcal{P}^1[\phi, \alpha](s) \stackrel{\text{def}}{=} \phi(s) \wedge \forall c : \alpha(c) \rightarrow \mathcal{R}[\phi(do(c, s))]$$

$$\mathcal{P}^n[\phi, \alpha](s) \stackrel{\text{def}}{=} \mathcal{P}^1[\mathcal{P}^{n-1}[\phi, \alpha], \alpha]$$

$$(\mathcal{P}^n[\phi, \alpha] \rightarrow \mathcal{P}^{n+1}[\phi, \alpha]) \Rightarrow (\mathcal{P}^n[\phi, \alpha] \equiv \mathcal{P}[\phi, \alpha])$$

This calculation can be done using *static domain reasoning* and provably terminates in several important cases.

Regressing Knowledge



Regressing Knowledge

It becomes possible to define the regression of our **Knows** macro:

$$
\begin{aligned}
&\mathcal{R}[\mathbf{Knows}(agt, \phi, do(c, s))] = \\
&\qquad [Obs(agt, c, s) = \{\} \rightarrow \mathbf{Knows}(agt, \phi, s)] \\
&\qquad \wedge [\exists o : Obs(agt, c, s) = o \wedge o \neq \{\} \rightarrow \\
&\qquad \mathbf{Knows}(agt, \forall c' : Obs(agt, c') = o \rightarrow \\
&\qquad\qquad \mathcal{R}[\mathcal{P}[\phi, PbU(agt)](do(c', s'))], s)]
\end{aligned}
$$

View-Based Reasoning

The regression operator can be modified to act over observation histories, instead of over situations:

$$\mathcal{R}[\mathbf{Knows}(agt, \phi, o \cdot h)] = \\ \mathbf{Knows}(agt, \forall c' : Obs(agt, c', s') = o \rightarrow \\ \mathcal{R}[\mathcal{P}[\phi, PbU(agt)](do(c', s'))], h)$$

We can equip agents with a situation calculus model of their own environment.

An Example

Ann and Bob have just received a party invitation.

We can prove the following:

$$\mathcal{D} \models \mathbf{Knows}(B, \neg\exists x : \mathbf{Knows}(A, partyAt(x)), S_0)$$

$$\mathcal{D} \models \neg\mathbf{Knows}(B, \neg\exists x : \mathbf{Knows}(A, partyAt(x)), do(leave(B), S_0))$$

$$\mathcal{D} \models \mathbf{Knows}(A, partyAt(C), do(read(A), S_0))$$

Summary (re-cap)

A robust account of **knowledge** based on observations, allowing for arbitrarily-long sequences of hidden actions.

- That subsumes existing accounts of knowledge
- With regression rules utilising the persistence condition
- Allowing agents to reason about their own knowledge using only their local information

Outline

Group-Level Knowledge

The basic group-level operator is "Everyone Knows":

$$\mathbf{EKnows}(G,\phi,s) \stackrel{\text{def}}{=} \bigwedge_{agt\in G} \mathbf{Knows}(agt,\phi,s)$$

$$\mathbf{EKnows}^2(G,\phi,s) \stackrel{\text{def}}{=} \mathbf{EKnows}(G,\mathbf{EKnows}(G,\phi),s)$$

$$\ldots$$

$$\mathbf{EKnows}^n(G,\phi,s) \stackrel{\text{def}}{=} \mathbf{EKnows}(G,\mathbf{EKnows}^{n-1}(G,\phi),s)$$

Eventually, we get "Common Knowledge":

$$\mathbf{CKnows}(G,\phi,s) \stackrel{\text{def}}{=} \mathbf{EKnows}^{\infty}(agt,\phi,s)$$

Regressing Group Knowledge

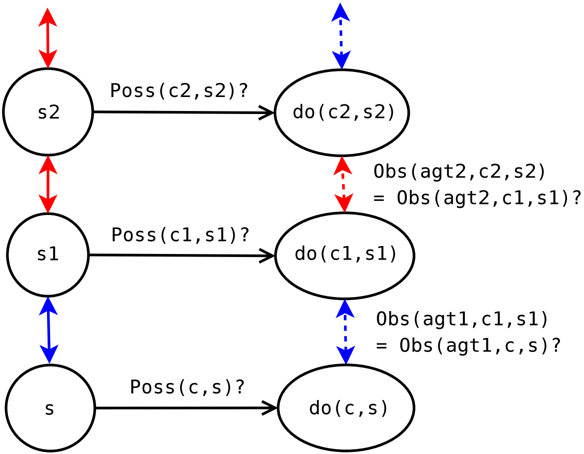
Since **EKnows** is finite, it can be expanded to perform regression.

CKnows is infinitary, so this won't work for common knowledge. We need to regress it directly. Maybe like this?

$$\mathcal{R}[\mathbf{CKnows}(G,\phi,do(c,s))] \stackrel{\text{def}}{=}$$
$$\exists o : \mathbf{CObs}(G,c,s) = o \,\wedge$$
$$\forall c' : \mathbf{CKnows}(G, Poss(c') \wedge \mathbf{CObs}(agt,c') = o \rightarrow \mathcal{R}[\phi[do(c',s)]], s)$$

It is impossible to express $\mathcal{R}[\mathbf{CKnows}]$ in terms of **CKnows**

Regressing Group Knowledge



Epistemic Path Language

$\mathcal{R}[\textbf{CKnows}]$ requires a more expressive epistemic language.

Dynamic Logics are formalisms for building programs from actions:

$$A\ ;\ ?\mathit{Poss}(B)\ ;\ B$$
$$A\ ;\ (B \cup C)$$
$$A^*\ ;\ ?\mathit{Done}$$
$$x := ?\ ;\ ?\mathit{Avail}(x)\ ;\ \mathit{pickup}(X)$$

But they don't *have* to be interpreted over actions.
More generally, DLs are logics of *paths*.

Epistemic Path Language

Idea from van Bentham, van Eijck and Kooi.
"Logics of Communication and Change", Info. & Comp., 2006

We can interpret Dynamic Logic epistemically:

$$\mathbf{KDo}(agt, s, s') \stackrel{\text{def}}{=} K(agt, s', s)$$

$$\mathbf{KDo}(?\phi, s, s') \stackrel{\text{def}}{=} s' = s \wedge \phi[s]$$

$$\mathbf{KDo}(\pi_1; \pi_2, s, s') \stackrel{\text{def}}{=} \exists s'' : \mathbf{KDo}(\pi_1, s, s'') \wedge \mathbf{KDo}(\pi_2, s'', s')$$

$$\mathbf{KDo}(\pi_1 \cup \pi_2, s, s') \stackrel{\text{def}}{=} \mathbf{KDo}(\pi_1, s, s'') \vee \mathbf{KDo}(\pi_2, s, s')$$

$$\mathbf{KDo}(\pi^*, s, s') \stackrel{\text{def}}{=} \text{refl.tran.closure}\left[\mathbf{KDo}(\pi, s, s'')\right]$$

Epistemic Path Language

New macro for path-based knowledge:

$$\mathbf{PKnows}(\pi, \phi, s) \stackrel{\text{def}}{=} \forall s' : \mathbf{KDo}(\pi, s, s') \rightarrow \phi[s']$$

Used like so:

$$\begin{aligned}
\mathbf{Knows}(agt, \phi, s) &\equiv \mathbf{PKnows}(agt, \phi, s) \\
\mathbf{Knows}(agt_1, \mathbf{Knows}(agt_2, \phi), s) &\equiv \mathbf{PKnows}(agt_1; agt_2, \phi, s) \\
\mathbf{EKnows}(G, \phi, s) &\equiv \mathbf{PKnows}(\bigcup_{agt \in G} agt, \phi, s) \\
\mathbf{CKnows}(G, \phi, s) &\equiv \mathbf{PKnows}((\bigcup_{agt \in G} agt)^*, \phi, s)
\end{aligned}$$

Regressing Epistemic Paths

It's now possible to formulate a regression rule for **PKnows** in synchronous domains:

$$\mathcal{R}[\mathbf{PKnows_0}(\pi, \phi, do(c, s))] \Rightarrow \forall c' : \mathbf{PKnows_0}(\mathcal{T}[\pi, c, c'], \mathcal{R}[\phi(c')], s)$$

$\mathcal{T}$ basically encodes the semantics of **KDo**

$$\mathcal{T}[agt] \stackrel{\text{def}}{=} \text{s.s.a. for } K \text{ fluent}$$

$$\mathcal{T}[?\phi] \stackrel{\text{def}}{=} ?\mathcal{R}[\phi]$$

$$\mathcal{T}[\pi_1 \cup \pi_2] \stackrel{\text{def}}{=} \mathcal{T}[\pi_1] \cup \mathcal{T}[\pi_2]$$

$$\mathcal{T}[\pi^*] \stackrel{\text{def}}{=} \mathcal{T}[\pi]^*$$

Asynchronicity

We can "fake" asynchronicity using $\mathbf{PKnows_0}$ and a stack of empty actions:

$$\mathcal{E}[do(c,s)] \stackrel{\text{def}}{=} do(\{\}, do(c, \mathcal{E}[s]))$$
$$\mathcal{E}^n[s] \stackrel{\text{def}}{=} \mathcal{E}[\mathcal{E}^{n-1}[s]]$$

Using a fixpoint construction that mirrors $\mathcal{P}$, define:

$$\mathbf{PKnows}(\pi, \phi, s) \stackrel{\text{def}}{=} \mathbf{PKnows_0}(\pi, \phi, \mathcal{E}^{\infty}[s])$$

We prove that $\mathbf{PKnows}(agt, \phi, s) \equiv \mathbf{Knows}(agt, \phi, s)$

An Example

Ann and Bob have just received a party invitation.

We can prove the following:

$$\mathcal{D} \models \neg \mathbf{PKnows}((A \cup B)^*, partyAt(C), S_0)$$

$$\mathcal{D} \models \mathbf{PKnows}((A \cup B)^*, \exists x : \mathbf{Knows}(B, partyAt(x)), do(read(B), S_0))$$

$$\mathcal{D} \models \mathbf{PKnows}((A \cup B)^*, partyAt(C)), do(read(A), do(read(B), S_0)))$$

Summary (re-cap)

Complex Epistemic Modalities: an encoding of group-level knowledge using the syntax of dynamic logic

- Built entirely use macro-expansion
- In which common knowledge is amenable to regression
- Incorporating arbitrarily-long sequences of hidden actions

Definition

In a *bisimulation* between two Kripke models $M = \langle S, V, R \rangle$ and $M' = \langle S, V', R' \rangle$, form a *relation* $\Re_B \subseteq S \times S'$ that satisfies the following properties,

Definition

In a *bisimulation* between two Kripke models $M = \langle S, V, R \rangle$ and $M' = \langle S, V', R' \rangle$, form a *relation* $\Re_B \subseteq S \times S'$ that satisfies the following properties,

1. $\Re_B$ satisfies *forward choice* if

   $$\forall s, t \in S$$
   $$\forall s' \in S'(\Re_B ss' \& (s,t) \in R) \Rightarrow \exists t' \in S'(\Re_B tt' \& (s',t') \in R')$$

Definition

In a *bisimulation* between two Kripke models $M = \langle S, V, R\rangle$ and $M' = \langle S, V', R'\rangle$, form a *relation* $\Re_B \subseteq S \times S'$ that satisfies the following properties,

1. $\Re_B$ satisfies *forward choice* if

   $\forall s, t \in S$
   $\forall s' \in S'(\Re_B ss' \& (s,t) \in R) \Rightarrow \exists t' \in S'(\Re_B tt' \& (s', t') \in R')$

2. $\Re_B$ satisfies *backwards choice* if

   $\forall s \in S$
   $\forall s', t' \in S'(\Re_B ss' \& (s', t') \in R') \Rightarrow \exists t \in S(\Re_B tt' \& (s, t) \in R)$

Definition

In a *bisimulation* between two Kripke models $M = \langle S, V, R \rangle$ and $M' = \langle S, V', R' \rangle$, form a *relation* $\Re_B \subseteq S \times S'$ that satisfies the following properties,

1. $\Re_B$ satisfies *forward choice* if

   $\forall s, t \in S$
   $\forall s' \in S'(\Re_B ss' \& (s, t) \in R) \Rightarrow \exists t' \in S'(\Re_B tt' \& (s', t') \in R')$

2. $\Re_B$ satisfies *backwards choice* if

   $\forall s \in S$
   $\forall s', t' \in S'(\Re_B ss' \& (s', t') \in R') \Rightarrow \exists t \in S(\Re_B tt' \& (s, t) \in R)$

3. Satisfies forward and backward choice, and
   $\forall s \in S \forall s' \in S'(\Re_B ss' \Rightarrow V(s) = V'(s'))$

Definition

A *zig zag morphism* is a bisimulation between two Kripke models $M = \langle S, V, R \rangle$ and $M' = \langle S, V', R' \rangle$, in terms of *relation* $\Re_B \subseteq S \times S'$ that satisfies forwards and backwards choice and additionally satisfies,

Definition

A *zig zag morphism* is a bisimulation between two Kripke models $M = \langle S, V, R \rangle$ and $M' = \langle S, V', R' \rangle$, in terms of *relation* $\Re_B \subseteq S \times S'$ that satisfies forwards and backwards choice and additionally satisfies,

- $domain(\Re_B) = S$ and $range(\Re_B) = S'$

Definition

A *zig zag morphism* is a bisimulation between two Kripke models $M = \langle S, V, R \rangle$ and $M' = \langle S, V', R' \rangle$, in terms of *relation* $\Re_B \subseteq S \times S'$ that satisfies forwards and backwards choice and additionally satisfies,

- $domain(\Re_B) = S$ and $range(\Re_B) = S'$

The *Zig-zag* idea

- Segerberg (1970)
- van Bentham (1983) and Plotkin and Stirling (1986)
- van der Hoek (1992)

Observations

Bisimulation is not canonical (it is *weaker* than graph isomorphism), example:

$$\begin{tikzcd}
 & \bullet v_2 \arrow[dr] & & \\
\bullet v_1 \arrow[ur] \arrow[dr] & & \bullet v_4 \arrow[r] & \bullet v_5 \\
 & \bullet v_3 \arrow[ur] & &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & \bullet v_3' \arrow[dr] & \\
\bullet v_1' \arrow[r] & \bullet v_2' \arrow[ur] \arrow[dr] & & \bullet v_5' \\
 & & \bullet v_4' \arrow[ur] &
\end{tikzcd}$$

G $\qquad$ G'

Observations

Bisimulation is not canonical (it is *weaker* than graph isomorphism), example:

$$\begin{tikzcd}
 & v_2 \arrow[dr] & & \\
v_1 \arrow[ur] \arrow[dr] & & v_4 \arrow[r] & v_5 \\
 & v_3 \arrow[ur] & &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & v'_3 \arrow[dr] & \\
v'_1 \arrow[r] & v'_2 \arrow[ur] \arrow[dr] & & v'_5 \\
 & & v'_4 \arrow[ur] &
\end{tikzcd}$$

G $\qquad\qquad$ G'

$$\Re_B = \{(v'_1, v_1), (v'_2, v_2), (v'_2, v_3), (v'_3, v_4), (v'_4, v_4), (v'_5, v_5)\}$$

Observations

Bisimulation is not canonical (it is *weaker* than graph isomorphism), example:

$$\begin{tikzcd}
& v_2 \arrow[dr] & & \\
v_1 \arrow[ur] \arrow[dr] & & v_4 \arrow[r] & v_5 \\
& v_3 \arrow[ur] & &
\end{tikzcd}
\qquad
\begin{tikzcd}
& & v_3' \arrow[dr] & \\
v_1' \arrow[r] & v_2' \arrow[ur] \arrow[dr] & & v_5' \\
& & v_4' \arrow[ur] &
\end{tikzcd}$$

G $\qquad$ G'

$$\Re_B = \{(v_1', v_1), (v_2', v_2), (v_2', v_3), (v_3', v_4), (v_4', v_4), (v_5', v_5)\}$$

Furthermore, for possible world structures, branching (sub-tree) bisimulation is NP-Complete (Dovier, 2003, subgraph bisimulation).

Observations

Calculus of communicating systems (CCS) (Milne, 1984)

Example: $a(b + c)$ $\quad\quad\quad\quad$ $a.b + a.c$

Observations

Calculus of communicating systems (CCS) (Milne, 1984)

Example: $a(b+c)$ $\qquad$ $a.b + a.c$

$$\begin{tikzcd} & & \bullet \\ \bullet \arrow[r, "a"] & \bullet \arrow[ur, "b"] \arrow[dr, "c"] & \\ & & \bullet \end{tikzcd}$$

G

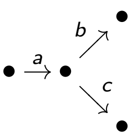
$$\begin{tikzcd} & \bullet \arrow[r, "b"] & {} \\ \bullet \arrow[ur, "a"] \arrow[dr, "a"] & & \\ & \bullet \arrow[r, "c"] & {} \end{tikzcd}$$

G'

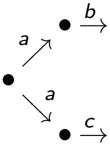
Observations

Calculus of communicating systems (CCS) (Milne, 1984)

Example: $a(b+c)$ $a.b+a.c$



G



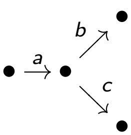
G'

τ transitions and branching time abstraction in bisimulation semantics (van Gabbek, 1996)

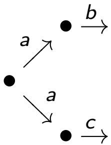
Observations

Calculus of communicating systems (CCS) (Milne, 1984)

Example: $a(b + c)$ $a.b + a.c$



G



G'

τ transitions and branching time abstraction in bisimulation semantics (van Gabbek, 1996)

However: generally utilised to establish correspondence based on *observed* computation histories, as opposed to correspondence of *forward* branching possible world structures.

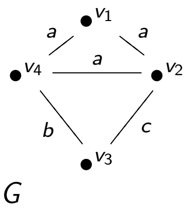
Graph isomorphism as an equivalence relation

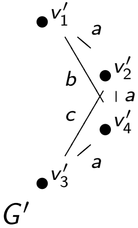
Let us define a relation $\Re_I$ on graphs by $G_1 \Re G_2$ if G_1 is isomorphic to G_2, $\Re_I$ is

- reflexive,
- symmetric and
- transitive,

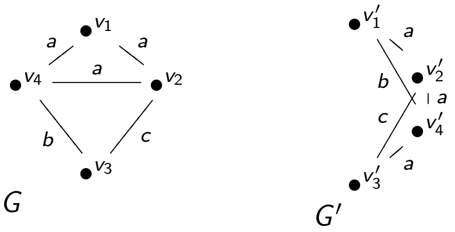
and therefore graph isomorphism is an *equivalence relation*.

Attributed graph isomorphism





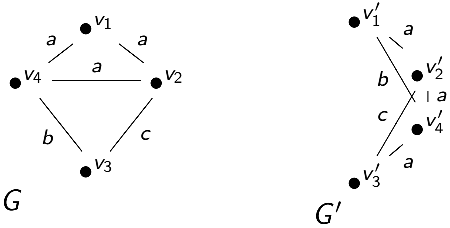
Attributed graph isomorphism



Bisimulation *could* solve this problem if (i) could differentiate between states and (ii) considered sufficiently long paths, for example the following bisimulation relation is impossible between graphs G_1 and G_2

$$\Re_B = \{(v_1', v_1), (v_2', v_2), (v_4', v_4), (v_1', v_1)\}$$

Attributed graph isomorphism



Bisimulation *could* solve this problem if (i) could differentiate between states and (ii) considered sufficiently long paths, for example the following bisimulation relation is impossible between graphs G_1 and G_2

$$\Re_B = \{(v_1', v_1), (v_2', v_2), (v_4', v_4), (v_1', v_1)\}$$

However, in general there are exponentially many paths (consistent with NP-complete result).

Publications

- Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. Rea- soning about Knowledge. The MIT Press, Cambridge, Massachesetts, 1995.
- Joseph Y. Halpern and Yoram Moses, Knowledge and Common Knowledge in a Distributed Environment, Journal of the ACM, Vol. 37, No. 3, pp. 549–587, 1990.
- Richard Scherl and Hector Levesque. Knowledge, Action, and the Frame Problem. Artificial Intelligence, 144:1-39, 2003.
- Ryan F. Kelly and Adrian R. Pearce. Knowledge and Observations in the Situation Calculus. In Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07), pages 841-843, 2007.

Publications (continued)

- Ryan F. Kelly and Adrian R. Pearce. Complex Epistemic Modalities in the Situation Calculus. In Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08), pages 611-620, 2008.
- Ryan Kelly. "Asynchronous Multi-Agent Reasoning in the Situation Calculus", PhD Thesis, The University of Melbourne, 2008

Summary