MULTIAGENT SYSTEMS 2E – CHAPTER 16
LOGICS FOR MULTI-AGENT SYSTEMS

Wiebe van der Hoek[1] and Michael Wooldridge[2]

[1] University of Liverpool, UK

[2] University of Oxford, UK

1 Overview

- The aim is to give an overview of the ways that theorists *conceptualise* agents, and to summarise some of the key developments in agent theory.
- Begin by answering the question: *why theory?*
- Discuss the various different *attitudes* that may be used to characterise agents.
- Introduce some problems associated with formalising attitudes.
- Introduce modal logic as a tool for reasoning about attitudes, focussing on knowledge/belief.
- Discuss Moore's theory of ability.
- Introduce the Cohen-Levesque theory of intention as a case study in agent theory.

2 Why Theory?

- Formal methods have (arguably) had little impact of general practice of software development: why should they be relevant in agent based systems?
- The answer is that we need to be able to give a *semantics* to the architectures, languages, and tools that we use — literally, a *meaning*.
- Without such a semantics, it is never clear exactly *what* is happening, or *why* it works.

- End users (e.g., programmers) need never read or understand these semantics, but progress cannot be made in language development until these semantics exist.
- In agent-based systems, we have a bag of concepts and tools, which are intuitively easy to understand (by means of metaphor and analogy), and have obvious potential.
- But we need theory to reach any kind of *profound* understanding of these tools.

3 Agents = Intentional Systems

- Where do theorists start from?
- The notion of an agent as an *intentional system*. . .
- So agent theorists start with the (strong) view of agents as intentional systems: one whose simplest consistent description requires the intentional stance.

4 Theories of Attitudes

- We want to be able to design and build computer systems in terms of ‘mentalistic’ notions.
- Before we can do this, we need to identify a tractable subset of these attitudes, and a model of how they interact to generate system behaviour.

- Some possibilities:

$$\text{information attitudes} \begin{cases} \text{belief} \\ \text{knowledge} \end{cases}$$

$$\text{pro-attitudes} \begin{cases} \text{desire} \\ \text{intention} \\ \text{obligation} \\ \text{commitment} \\ \text{choice} \\ \ldots \end{cases}$$

5 Formalising Attitudes

- So how do we formalise attitudes?
- Consider. . .

  Janine believes Cronos is father of Zeus.

- Naive translation into first-order logic:

$$Bel(Janine, Father(Zeus, Cronos))$$

- But. . .
  - the second argument to the Bel predicate is a *formula* of first-order logic, not a term;
    need to be able to apply 'Bel' to formulae;
  - allows us to substitute terms with the same denotation: consider $(Zeus = Jupiter)$
    intentional notions are referentially opaque.

- So, there are two sorts of problems to be addressed in develping a logical formalism for intentional notions:
  - a *syntactic* one (intentional notions refer to sentences); and
  - a *semantic* one (no substitution of equivalents).
- Thus any formalism can be characterized in terms of two attributes: its *language of formulation*, and *semantic model*:
- Two fundamental approaches to the syntactic problem:
  - use a *modal* language, which contains *modal operators*, which are applied to formulae;
  - use a *meta-language*: a first-order language containing terms that denote formulae of some other *object-language*.
- We will focus on modal languages, and in particular, *normal modal logics*, with *possible worlds semantics*.

6 Normal Modal Logic for Knowledge

- Syntax is classical propositional logic, plus an operator K for 'knows that'.

  Vocabulary:

$\Phi = \{p, q, r, \ldots\}$	primitive propositions
$\wedge, \vee, \neg, \ldots$	classical connectives
K	modal connective

  Syntax:

$$
\begin{array}{rcl}
\langle \mathit{wff} \rangle & ::= & \text{any member of } \Phi \\
& \mid & \neg \langle \mathit{wff} \rangle \\
& \mid & \langle \mathit{wff} \rangle \vee \langle \mathit{wff} \rangle \\
& \mid & K \langle \mathit{wff} \rangle
\end{array}
$$

- Example formulae:

$$K(p \wedge q)$$
$$K(p \wedge Kq)$$

- Semantics are trickier. The idea is that an agent's beliefs can be characterized as a set of *possible worlds*, in the following way.
- Consider an agent playing a card game such as poker, who possessed the ace of spades.

  How could she deduce what cards were held by her opponents?
- First calculate all the various ways that the cards in the pack could possibly have been distributed among the various players.
- The systematically eliminate all those configurations which are *not possible, given what she knows*.

  (For example, any configuration in which she did not possess the ace of spades could be rejected.)

- Each configuration remaining after this is a *world*; a state of affairs considered possible, given what she knows.
- Something true in *all* our agent's possibilities is believed by the agent.

  For example, in all our agent's *epistemic alternatives*, she has the ace of spades.
- Two advantages:
  - remains neutral on the cognitive structure of agents;
  - the associated mathematical theory is very nice!

- To formalise all this, let W be a set of worlds, and let $R \subseteq W \times W$ be a binary relation on W, characterising what worlds the agent considers possible.
- For example, if $(w, w') \in R$, then if the agent was *actually* in world w, then as far as it was concerned, it *might* be in world w'.
- Semantics of formulae are given relative to worlds: in particular:

  $K\phi$ is true in world w iff ϕ is true in all worlds w' such that $(w, w') \in R$.

- Two basic properties of this definition:
  - the following axiom schema is valid:
    $K(\phi \Rightarrow \psi) \Rightarrow (K\phi \Rightarrow K\psi)$
  - if ϕ is valid, then $K\phi$ is valid.
- Thus *agent's knowledge is closed under logical consequence*: this is *logical omniscience*.

  This is *not* a desirable property!

- The most interesting properties of this logic turn out to be those relating to the properties we can impose on accessibility relation R.

  By imposing various constraints, we end up getting out various axioms; there are *lots* of these, but the most important are:

$$\begin{array}{ll} \mathrm{T} & K\phi \Rightarrow \phi \\ \mathrm{D} & K\phi \Rightarrow \neg K \neg \phi \\ 4 & K\phi \Rightarrow KK\phi \\ 5 & \neg K\phi \Rightarrow K\neg K\phi. \end{array}$$

Interpreting the Axioms

- Axiom T is the *knowledge axiom*: it says that what is known is true.
- Axiom D is the *consistency axiom*: if you know ϕ, you can't also know $\neg\phi$.
- Axiom 4 is *positive introspection*: if you know ϕ, you know you know ϕ.
- Axiom 5 is *negative introspection*: you are aware of what you don't know.

Systems of Knowledge & Belief

- We can (to a certain extent) pick and choose which axioms we want to represent our agents.
- All of these (KTD45) constitute the logical system S5.

  Often chosen as a logic of *idealised knowledge*.
- S5 without T is weak-S5, or KD45.

  Often chosen as a logic of *idealised belief*.

7 Knowledge & Action

- Most-studied aspect of practical reasoning agents:

  interaction between knowledge and action.

- Moore's 1977 analysis is best-known in this area.
- Formal tools:
  - a modal logic with Kripke semantics + dynamic logic-style representation for action;
  - *but* showed how Kripke semantics could be axiomatized in a first-order meta-language;
  - modal formulae then translated to meta-language using axiomatization;
  - modal theorem proving reduces to meta-language theorem proving.

- Moore considered 2 aspects of interaction between knowledge and action:

  1. As a result of performing an action, an agent can gain knowledge.

     Agents can perform "test" actions, in order to find things out.
  2. In order to perform some actions, an agent needs knowledge: these are *knowledge pre-conditions*.

     For example, in order to open a safe, it is necessary to know the combination.

- Culminated in defn of *ability*: what it means to be able to do bring something about.

- Axiomatising standard logical connectives:

$$\begin{array}{l}
\forall w.\mathit{True}(w, \ulcorner \neg\phi \urcorner) \Leftrightarrow \neg\mathit{True}(w, \ulcorner \phi \urcorner) \\
\forall w.\mathit{True}(w, \ulcorner \phi \wedge \psi \urcorner) \Leftrightarrow \mathit{True}(w, \ulcorner \phi \urcorner) \wedge \mathit{True}(w, \ulcorner \psi \urcorner) \\
\forall w.\mathit{True}(w, \ulcorner \phi \vee \psi \urcorner) \Leftrightarrow \mathit{True}(w, \ulcorner \phi \urcorner) \vee \mathit{True}(w, \ulcorner \psi \urcorner) \\
\forall w.\mathit{True}(w, \ulcorner \phi \Rightarrow \psi \urcorner) \Leftrightarrow \mathit{True}(w, \ulcorner \phi \urcorner) \Rightarrow \mathit{True}(w, \ulcorner \psi \urcorner) \\
\forall w.\mathit{True}(w, \ulcorner \phi \Leftrightarrow \psi \urcorner) \Leftrightarrow (\mathit{True}(w, \ulcorner \phi \urcorner) \Leftrightarrow \mathit{True}(w, \ulcorner \psi \urcorner))
\end{array}$$

Here, True is a meta-language predicate:

– 1st argument is a term denoting a world;

– 2nd argument a term denoting modal language formula.

Frege quotes, $\ulcorner\ \urcorner$, used to quote modal language formula.

- Axiomatizing the knowledge connective: basic possible world semantics:

$$\forall w \cdot \mathit{True}(w, \ulcorner(\mathsf{Know}\phi)\urcorner) \Leftrightarrow \forall w' \cdot K(w, w') \Rightarrow \mathit{True}(w', \ulcorner\phi\urcorner)$$

  Here, K is a meta-language predicate used to represent the knowledge accessibility relation.

- Other axioms added to represent properties of knowledge.

  Reflexive: $\forall w.K(w, w)$

  Transitive: $\forall w, w', w'' \cdot K(w, w') \wedge K(w', w'') \Rightarrow K(w, w'')$

  Euclidean: $\forall w, w', w'' \cdot K(w, w') \wedge K(w'', w') \Rightarrow K(w, w'')$

  Ensures that K is *equivalence relation*.

- Now we need some apparatus for representing *actions*.
- Add a meta-language predicate $R(a, w, w')$ to mean that w' is a world that could result from performing action a in world w.
- Then introduce a modal operator $(\mathsf{Res}\ a\ \phi)$ to mean that *after action a is performed, ϕ will be true.*

$$\begin{array}{l}\forall w.True(w, \ulcorner(\mathsf{Res}\ a\ \phi)\urcorner) \Leftrightarrow \\ \quad \exists w' \cdot R(a, w, w') \wedge \forall w'' \cdot R(a, w, w'') \Rightarrow True(w'', \ulcorner\phi\urcorner)\end{array}$$

  – first conjunct says the action is *possible*;
  – second says that a neccesary consequence of performing action is ϕ.

- Now we can define ability, via modal Can operator.

$$\begin{aligned}\forall w \cdot \mathit{True}(w, \ulcorner(\mathsf{Can}\ \phi)\urcorner) &\Leftrightarrow \\ &\exists a.\mathit{True}(w, \ulcorner(\mathsf{Know}\ (\mathsf{Res}\ a\ \phi))\urcorner)\end{aligned}$$

  So agent can achieve ϕ if there exists some action a, such that agent knows that the result of performing a is ϕ.

- Note the way a is quantified w.r.t. the Know modality.

  Implies agent knows the identity of the action.

  Has a "definite description" of it.

  (Terminology: a is quantified *de re*.)

- We can weaken the definition, to capture the case where an agent performs an action to find out how to achieve goal.

$$\begin{aligned}\forall w \cdot \mathit{True}(w, \ulcorner(\mathsf{Can}\ \phi)\urcorner) \Leftrightarrow \\ \exists a.\mathit{True}(w, \ulcorner(\mathsf{Know}\ (\mathsf{Res}\ a\ \phi))\urcorner) \vee \\ \exists a.\mathit{True}(w, \ulcorner(\mathsf{Know}\ (\mathsf{Res}\ a\ (\mathsf{Can}\ \phi)))\urcorner)\end{aligned}$$

A circular definition?

No, interpret as a *fixed point*.

- Critique of Moore's formaism:

  1. Translating modal language into a first-order one and then theorem proving in first-order language is inefficient.

     "Hard-wired" modal theorem provers will be more efficient.

  2. Formulae resulting from the translation process are complicated and unintuitive.

     Original structure (and hence sense) is lost.

  3. Moore's formalism based on possible worlds: falls prey to logical omniscience.

     Definition of ability is somewhat vacuous.

- But probably first serious attempt to use tools of mathematical logic (incl. modal & dynamic logic) to bear on rational agency.

8 Intention

- We have *one aspect* of an agent, but knowledge/belief alone does not completely characterise an agents.
- We need a *set* of connectives, for talking about an agent's *pro-attitudes* as well.
- Agent needs to achieve a *rational balance* between its attitudes:
  - should not be *over-committed*;
  - should not be *under-committed*.
- Here, we review one attempt to produce a coherent account of how the components of an agent's cognitive state hold together: the theory of intention developed by Cohen & Levesque.
- Here we mean intention as in. . .

It is my intention to prepare my slides.

8.1 What is intention?

- Two sorts:
  - *present directed*
    - attitude to an action
    - function causally in producing behaviour.
  - *future directed*
    - attitude to a proposition
    - serve to coordinate future activity.
- We are here concerned with *future directed* intentions.

Following Bratman (1987) Cohen-Levesque identify seven properties that must be satisfied by intention:

1. Intentions pose problems for agents, who need to determine ways of achieving them.

   If I have an intention to ϕ, you would expect me to devote resources to deciding how to bring about ϕ.

2. Intentions provide a 'filter' for adopting other intentions, which must not conflict.

   If I have an intention to ϕ, you would expect me to adopt an intention ψ such that ϕ and ψ are mutually exclusive.

3. Agents track the success of their intentions, and are inclined to try again if their attempts fail.

   If an agent's first attempt to achieve ϕ fails, then all other things being equal, it will try an alternative plan to achieve ϕ.

In addition. . .

- Agents believe their intentions are possible.

  That is, they believe there is at least some way that the intentions could be brought about. (CTL notation:* $\mathsf{E}\Diamond\phi$*).*

- Agents do not believe they will not bring about their intentions.

  It would not be rational of me to adopt an intention to ϕ *if I believed* ϕ *was not possible. (CTL* notation:* $\mathsf{A}\Box\neg\phi$*.)*

- Under certain circumstances, agents believe they will bring about their intentions.

  It would not normally be rational of me to believe that I would *bring my intentions about; intentions can* fail*. Moreover, it does not make sense that if I believe ϕ is inevitable (CTL*: $\mathsf{A}\Diamond\phi$) that I would adopt it as an intention.*

- Agents need not intend all the expected side effects of their intentions.

  If I believe $\phi \Rightarrow \psi$ and I intend that ϕ, I do not necessarily intend ψ also. (Intentions are not closed under implication.)

  This last problem is known as the *dentist* problem. I may believe that going to the dentist involves pain, and I may also intend to go to the dentist — but this does not imply that I intend to suffer pain!

- Cohen-Levesque use a *multi-modal logic* with the following major constructs:

$(\mathsf{Bel}\ x\ \phi)$	x believes ϕ
$(\mathsf{Goal}\ x\ \phi)$	x has goal of ϕ
$(\mathsf{Happens}\ \alpha)$	action α happens next
$(\mathsf{Done}\ \alpha)$	action α has just happened

- Semantics are possible worlds.
- Each world is infinitely long linear sequence of states.

- Each agent allocated:
  - *belief accessibility relation* — B
    for every agent/time pair, gives a set of belief accessible worlds;
    Euclidean, serial, transitive — gives belief logic KD45.
  - *goal accessibility relation* — G
    for every agent/time pair, gives a set of goal accessible worlds.
    Serial — gives goal logic KD.

- A constraint: $G \subseteq B$.
  - Gives the following inter-modal validity:

$$\models (\mathsf{Bel}\ i\ \phi) \Rightarrow (\mathsf{Goal}\ i\ \phi)$$

  - A *realism* property — agents *accept the inevitable*.
- Another constraint:

$$\models (\mathsf{Goal}\ i\ \phi) \Rightarrow \Diamond\neg(\mathsf{Goal}\ i\ \phi)$$

C&L claim this assumption captures following properties:

  - agents do not persist with goals forever;
  - agents do not indefinitely defer working on goals.

- Add in some operators for describing the structure of event sequences

  $\alpha; \alpha'$ α followed by α'

  $\alpha?$ 'test action' α

- Also add some operators of temporal logic, "$\Box$" (always), and "$\Diamond$" (sometime) can be defined as abbreviations, along with a "strict" sometime operator, Later:

$$
\begin{aligned}
\Diamond \alpha &\hat{=} \exists x \cdot (\mathsf{Happens}\ x; \alpha?) \\
\Box \alpha &\hat{=} \neg \Diamond \neg \alpha \\
(\mathsf{Later}\ p) &\hat{=} \neg p \wedge \Diamond p
\end{aligned}
$$

- Finally, a temporal precedence operator, $(\mathsf{Before}\ p\ q)$.
- First major derived construct is a *persistent* goal.

$$
\begin{array}{ll}
(\mathsf{P-Goal}\ x\ p) \hat{=} & \\
\quad (\mathsf{Goal}\ x\ (\mathsf{Later}\ p)) & \wedge \\
\quad (\mathsf{Bel}\ x\ \neg p) & \wedge \\
\quad \left[\begin{array}{l} \mathsf{Before} \\ \quad ((\mathsf{Bel}\ x\ p) \vee (\mathsf{Bel}\ x\ \Box \neg p)) \\ \quad \neg(\mathsf{Goal}\ x\ (\mathsf{Later}\ p)) \end{array}\right] &
\end{array}
$$

- So, an agent has a persistent goal of p if:

  1. It has a goal that p eventually becomes true, and believes that p is not currently true.
  2. Before it drops the goal, one of the following conditions must hold:
     – the agent believes the goal has been satisfied;
     – the agent believes the goal will never be satisfied.

- Next, intention:

$$
\begin{array}{l}
(\mathsf{Intend}\ x\ \alpha) \hat{=} \\
\quad (\mathsf{P} - \mathsf{Goal}\ x \\
\qquad [\mathsf{Done}\ x\ (\mathsf{Bel}\ x\ (\mathsf{Happens}\ \alpha))?; \alpha] \\
\quad)
\end{array}
$$

- So, an agent has an intention to do α if: it has a persistent goal to have believed it was about to do α, and then done α.
- C&L discuss how this definition satisfies desiderata for intention.
- Main point: avoids *ever commitment*.
- Adaptation of definition allows for *relativised intentions*. Example: *I have an intention to prepare slides for the tutorial,* relative *to the belief that I will be paid for tutorial. If I ever come to believe that I will not be paid, the intention evaporates. . .*

- Critique of C&L theory of intention (Singh, 1992):
  - does not capture and adequate notion of “competence”;
  - does not adequately represent intentions to do composite actions;
  - requires that agents know what they are about to do — fully elaborated intentions;
  - disallows multiple intentions.

9 Semantics for Speech Acts

- C&L used their theory of intention to develop a theory of several speech acts.
- Key observation: illocutionary acts are *complex event types* (cf. actions).
- C&L use their dynamic logic-style formalism for representing these actions.
- We will look at *request*.

- First, define *alternating belief*.

$$(\mathsf{AltBel}\ n\ x\ y\ p) \hat{=} \underbrace{(\mathsf{Bel}\ x\ (\mathsf{Bel}\ y\ (\mathsf{Bel}\ x\ \cdots (\mathsf{Bel}\ x}_{n \text{ times}}\ p\ \underbrace{)\cdots)}_{n \text{ times}}$$

- And the related concept of *mutual belief*.

$$(\mathsf{M}-\mathsf{Bel}\ x\ y\ p) \hat{=} \forall n \cdot (\mathsf{AltBel}\ n\ x\ y\ p)$$

- An *attempt* is defined as a complex action expression.

  (Hence the use of curly brackets, to distinguish from predicate or modal operator.)

$$\{\text{Attempt } x\ e\ p\ q\} \hat{=} \left[\begin{array}{ll} (\text{Bel } x\ \neg p) & \wedge \\ (\text{Goal } x\ (\text{Happens } x\ e; p?)) & \wedge \\ (\text{Intend } x\ e; q?) & \end{array}\right]?; e$$

- In English:

  "An attempt is a complex action that agents perform when they do something (e) desiring to bring about some effect (p) but with intent to produce at least some result (q)".

  Here:

  – p represents ultimate goal that agent is aiming for by doing e;

  – proposition q represents what it takes to at least make an "honest effort" to achieve p.

- Definition of *helpfulness* needed:

$$
\begin{array}{l}
(\mathsf{Helpful}\ x\ y) \hat{=} \\
\quad \forall e \cdot \left[\begin{array}{l} (\mathsf{Bel}\ x\ (\mathsf{Goal}\ y\ \Diamond(\mathsf{Done}\ x\ e))) \ \wedge \\ \neg(\mathsf{Goal}\ x\ \Box\neg(\mathsf{Done}\ x\ e)) \end{array} \right] \\
\qquad \Rightarrow (\mathsf{Goal}\ x\ \Diamond(\mathsf{Done}\ x\ e))
\end{array}
$$

- In English:

  "[C]onsider an agent [x] to be helpful to another agent [y] if, for any action [e] he adopts the other agent's goal that he eventually do that action, whenever such a goal would not conflict with his own".

- Definition of *requests*:

$$
\begin{array}{l}
\{\textsf{Request}\ \mathit{spkr}\ \mathit{addr}\ e\ \alpha\} \hat{=} \\
\qquad \{\textsf{Attempt}\ \mathit{spkr}\ e\ \phi \\
\qquad\quad (\textsf{M}-\textsf{Bel}\ \mathit{addr}\ \mathit{spkr}\ (\textsf{Goal}\ \mathit{spkr}\ \phi)) \\
\qquad \}
\end{array}
$$

where ϕ is

$$
\begin{array}{l}
\Diamond(\textsf{Done}\ \mathit{addr}\ \alpha) \wedge \\
(\textsf{Intend}\ \mathit{addr}\ \alpha \\
\quad \left[\begin{array}{l} (\textsf{Goal}\ \mathit{spkr}\ \Diamond(\textsf{Done}\ \mathit{addr}\ \alpha)) \wedge \\ (\textsf{Helpful}\ \mathit{addr}\ \mathit{spkr}) \end{array}\right] \\
)
\end{array}
$$

- In English:

  A request is an attempt on the part of $spkr$, by doing e, to bring about a state where, ideally, 1) $addr$ intends α, (relative to the $spkr$ still having that goal, and $addr$ still being helpfully inclined to $spkr$), and 2) $addr$ actually eventually does α, or at least brings about a state where $addr$ believes it is mutually believed that it wants the ideal situation.

- By this definition, there is no primitive request act:

  "[A] speaker is viewed as having performed a request if he executes any sequence of actions that produces the needed effects".

Game Logics

- introduce ATL – a *cooperation logic*;
- show you why ATL is important;
- discuss *reasoning* with ATL
- ATL + knowledge.

ATL: A Logic for Multiagent Systems

- *Alternating-time Temporal Logic* ("ATL") introduced in 1997 for reasoning about *game-like distributed systems*.
- Main item of novelty: allows us to talk about *powers* of system components.
- Generalises *Computation Tree Logic* (CTL).

Cooperation Modalities in ATL

- Basic expression in ATL is the *cooperation modality*:

$$\langle\langle C \rangle\rangle \phi$$

means

"coalition C can cooperate to ensure that ϕ"

- Cooperation modalities combined with *temporal* modalities:

$\bigcirc$	in the next state
$\diamondsuit$	eventually
$\square$	always
$\mathcal{U}$	until

Coalition logic is the "$\bigcirc$"-fragment.

Example ATL Formulae

$$\langle\langle gb, tb \rangle\rangle \Diamond peace$$

George Bush and Tony Blair can cooperate to ensure that, eventually, there is peace.

Note:

- does not imply that gb and tb *know what the strategy is*,
- nor that they *choose* this strategy!

It just says: *there is in principle some way that they could ensure there is eventually peace*.

Why is ATL Important?

Consider the mechanism requirements:

> *Two agents, A and B, must choose between two outcomes, p and q. We want a mechanism that will allow them to choose, which will satisfy the following requirements. First, whatever happens, we definitely want an outcome to result — that is, we want either p or q to be selected. Second, we really do want the agents to be able to collectively choose an outcome. However, we do not want them to be able to bring about both outcomes simultaneously. Similarly, we do not want either agent to dominate: we want them both to have equal power.*

Why is ATL Important?

Pauly realised you could express these requirements in ATL:

$$\langle\langle A, B\rangle\rangle \bigcirc p$$
$$\langle\langle A, B\rangle\rangle \bigcirc q$$
$$\langle\langle\rangle\rangle \bigcirc (p \vee q)$$
$$\langle\langle\rangle\rangle \bigcirc \neg(p \wedge q)$$
$$\neg\langle\langle A\rangle\rangle \bigcirc p$$
$$\neg\langle\langle B\rangle\rangle \bigcirc p$$
$$\neg\langle\langle A\rangle\rangle \bigcirc q$$
$$\neg\langle\langle B\rangle\rangle \bigcirc q$$

. . . you can then *automatically verify* whether a particular mechanism satisfies these requirements.

(Pauly & Wooldridge, *GTDT*, 2003)

Other Properties Specified Using ATL

Capture properties of *qualitative coalitional games* (QCGs):
$SUCC(C)$ means that C are *successful*:

$$SUCC(C) \equiv \langle\langle C \rangle\rangle \bigwedge_{i \in C} goal_i$$

Other Properties Specified Using ATL

$VETO(i, j)$ means "agent i is a veto player for agent j":

$$VETO(i, j) \equiv \bigwedge_{C} \langle\langle C \rangle\rangle \Diamond goal_j \Rightarrow \neg \langle\langle C \setminus \{i\} \rangle\rangle \Diamond goal_j$$

Other Properties Specified Using ATL

And so capture *mutual dependencies* between agents.

$$MD(C) \equiv \bigwedge_{i,j \in C} VETO(i,j)$$

(Wooldridge & Dunne, *AIJ*, 2004)

Reasoning with ATL: Satisfiability

- The *satisfiability problem* for ATL is as follows:

  given ATL formula ϕ is there some way that ϕ could be true?

- So: how hard is satisfiability in ATL?
  - For *Coalition Logic*: *PSPACE-complete* (Pauly, 2001).
  - For *CTL*: *EXPTIME-complete* – a lower bound for ATL.

Reasoning with ATL: Satisfiability

- First result:

  For *any fixed set of agents* Ag, satisfiability for ATL formulae over Ag is EXPTIME-complete.
  (van Drimmelen, 2003).

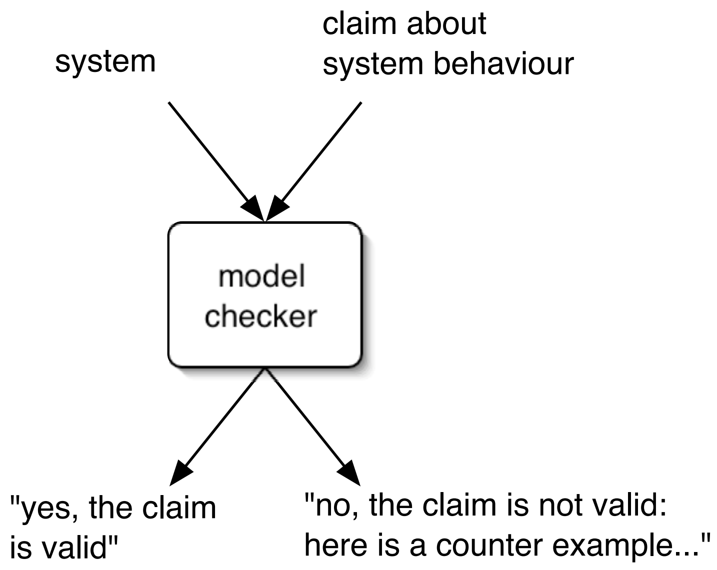
  But if the set of agents is not fixed, van Drimmelen's algorithm only works in 2EXPTIME... the *general problem* is left open.

- Second result:

  The satisfiability problem for *arbitrary* ATL formulae is EXPTIME-complete (and hence no harder than CTL).
  (Walther, Lutz, Wolter, Wooldridge, 2006)

Reasoning with ATL: Model-checking

Most work on verification uses *model checking*.



A *system/model* (S) for ATL is a *state transition graph*.

Notation:

$$S \models \phi$$

Reasoning with ATL: Model-checking

- But *how do we represent the model*? This affects the complexity.
- *If the model defined as state transition graph*

  . . . the problem can be solved in time polynomial in the size of the state transition graph.

  This is *not* a feasible representation:

  State transition graph is exponential in the number of system variables.

Reasoning with ATL: Model Checking

- *Practical* model checkers use *high-level* model specification languages.
- *Reactive modules*: a rule-based language for model specification.

  Allows *succinct* specification of large models.

  Essentially a guarded command language.

Reasoning with ATL: Model Checking

- How complex is model checking for this representation?
- *Exactly as hard as theorem proving in the corresponding language*:

ATL	EXPTIME-complete
Coalition Logic	PSPACE-complete
propositional logic	co-NP-complete

(van der Hoek, Lomuscio, Wooldridge, AAMAS06)

Knowledge + ATL

- Adding *knowledge* allows us to talk about *the ability to communicate*,

$$\langle\langle i \rangle\rangle \Diamond K_j p$$

. . . as well as *knowledge-pre-conditions*:

$$(\langle\langle i \rangle\rangle \bigcirc open\_safe) \Leftrightarrow K_i key$$

(van der Hoek & Wooldridge, *Studia Logica*, 2003)

- But do we require that agents *know* the strategy?
  Surprisingly difficult to get right, semantically.
  (See work of Jamroga & van der Hoek, Herzig et al, Agotnes.)