On Joint Ability in the Presence of Sensing

Hojjat Ghaderi and Hector Levesque
Department of Computer Science
University of Toronto
Toronto, ON M5S 3G4, Canada
{hojjat,hector}@cs.toronto.edu

Yves Lespérance
Department of Computer Science and Engineering
York University
Toronto, ON M3J 1P3, Canada
lesperan@cse.yorku.ca

Abstract

A central problem in the analysis of teams of agents is to assess when a group of autonomous agents, who may have private beliefs and goals, know enough together to be able to achieve a goal, should they so desire. In this paper, we present a definition of joint ability in the presence of sensing. We show through some simple examples involving private and public actions that it makes appropriate predictions with respect to coordination.

1 Introduction

An individual agent can often achieve a goal even if he does not initially know all the steps to follow, as long as he can *sense* along the way enough information to know what to do next, until the goal is attained. Clearly, one may delegate a goal only to an agent that is able to achieve it. Moore [1985] and others [Davis, 1994; Lespérance *et al.*, 2000] developed logical accounts of single agent *ability* in this sense.

Given this, an obvious question is how to extend this notion to *teams* of agents: when does a group know enough together, despite any incomplete knowledge or even false beliefs that they may have about the world or each other, to be *jointly able* to achieve a goal. Crucially, the agents need to know enough to stay coordinated. Unlike in the single-agent case, the mere existence of a joint plan mutually believed to achieve the goal (as in [Wooldridge and Jennings, 1999]) is not sufficient, since there may be several incompatible working plans and the agents may not be able to choose a share that coordinates with those of the others.

The issue of coordination has been thoroughly explored in game theory [Osborne and Rubinstein, 1999]. However, a major limitation of the classical game theoretic framework is that it assumes that there is a *complete specification* of the structure of the game including the beliefs of the agents. It is often also assumed that this structure is common knowledge among the agents. Recent work on the symbolic logic of games allows more incomplete and qualitative specifications, and supports symbolic reasoning over very large state spaces. However, most of this work, such as Coalition Logic [Pauly, 2002] and ATEL [van der Hoek and Wooldridge, 2003], is propositional, which limits expressiveness. More importantly, it ignores the issue of *coordination* within a coalition and simply assumes that a team can achieve a goal if there exists a strategy profile (or joint plan) over the agents that achieves it. This is only sufficient if we assume that the agents can communicate arbitrarily to coordinate as needed.

Ghaderi *et al.* [2007] proposed a logical framework to model the coordination of teams of agents based on the situation calculus. Their formalization avoids both of the limitations mentioned above: it supports reasoning on the basis of very incomplete specifications about the belief states of the agents, and it does not trivialize the issue of coordination. The formalization involves iterated elimination of dominated strategies [Osborne and Rubinstein, 1999]. Each agent eliminates strategies that are not as good as others given her private beliefs about the world and about what strategies the other agents would eliminate. This elimination process is repeated until it converges to a set of *preferred strategies* for each agent. Joint ability is said to hold if all combinations of preferred strategies succeed in achieving the goal.

However, Ghaderi *et al.* only considered example domains involving ordinary "world changing" actions. In this paper, we extend their account and show that it correctly handles domains with *sensing actions*, actions that allow an agent to obtain information by observing some aspect of the environment. We use an account of sensing actions and their effects on knowledge from [Shapiro *et al.*, 1998] and show that the techniques proposed by Ghaderi *et al.* to prove joint ability or the lack of it can be generalized to deal with sensing actions. This is significant, as it requires dealing with knowledge change, and strategies that may branch depending on what is learned by sensing. The space of strategies quickly becomes extremely large and it is significant that our symbolic proof techniques nonetheless allow results to be proven, even with only incomplete specifications of the agents' knowledge. Note that our approach can be also used to handle basic "informing" communication actions where the value of a fluent is communicated by one agent to another. These actions work very much like sensing, the difference being that it is the recipient of the communication that gets the new information.

In the next section, we describe a simple setup to test these ideas involving a safe with two locks. We then present the formalization of this domain and the definition of joint ability in the situation calculus. We then show the kind of ability results we can obtain in this formalization. Finally, we summarize our contributions and discuss future work.

2 Opening A Safe Together

Perhaps the simplest non-trivial example of ability in the single agent case was presented by Moore [1985]. He presents the example of a safe that can be opened by dialing the correct combination. Imagine that the safe explodes or jams or turns on a security alarm when any other number is dialed. Suppose there is an agent who does not know the combination of the safe. Although a plan surely exists to open the safe and the agent knows this, we would say that the agent is not able to open the safe. But if the correct combination is written on a piece of paper that the agent can pick up and read, we would now say that the agent is able to open the safe. This, in its most basic form, shows how ability depends on knowledge which itself depends on the available sensing actions.

To illustrate our formalization of joint ability, we will use a series of examples based on a two-person safe. The idea, roughly, is that two combinations are needed to open the safe, so that P and Q may be able to open the safe together even though neither one may know enough to do it alone. We also want to consider variants where, for example, Q knows both combinations allowing him to open the safe alone if P does not interfere, although P might not see this and ruin the plan.

To focus on the essential details only, we make a few simplifications. First of all, safe combinations will be binary: the safe has two locks A and B, and each lock has two buttons, 0 and 1. To open the safe, the correct button for lock A must first be pushed (using action *pA0* or *pA1*) – this puts the safe on standby – and then the correct button for lock B must be pushed (using action *pB0* or *pB1*). Any button pushing other than this sequence sets off an alarm, and the game is lost. Instead of having combinations written on a pieces of paper, we assume that there are two binary sensing actions, *sA* and *sB*, that cause the agent performing them to come to know the combination of the lock in question. We also assume that the agents act synchronously and in turn: P acts first and then they alternate. The goal in all cases will be to open the safe in exactly 4 steps without activating the alarm. We consider four variants of this setup with different assumptions.

Example 1: Suppose nothing is specified about the agents knowledge about the correct combinations of the locks. Actions are restricted in such a way that P and Q can only choose among actions $\{pA0, pA1, sA\}$ and $\{pB0, pB1, sB\}$, respectively. The actions are public so each agent gets to see the actions of the other agent. For this example, we want to say the agents can jointly open the safe. If the agents know nothing, the intuitive joint plan would look like this: P senses the combination of A, Q senses the combination of B, P pushes the correct button for A, and Q pushes the correct button for B. Note that if agents have extra information, e.g. P knows in advance the combination of A, other successful joint plans will exist, but as we will see, they do not cause any coordination problem.

Example 2: Suppose everything is exactly as in example 1 except that *sA* does not provide any information (the sensor is broken). In this case, we want to say that there is not enough information to conclude that the agents can open the safe. In fact, we will show that if P does not know the combination of lock A, they are provably not jointly able to open the safe.

Example 3: Suppose everything is as in example 1 except that the actions are not public, so the agents do not see what actions the other agent has performed (each just knows that some action has been performed by the other, so there is no confusion about whose turn it is). We will show that in this case seeing the other agent's actions is not necessary and that the agents are jointly able to open the safe, again without any need for extra assumptions about the beliefs of the agents.

Example 4: Suppose everything is as in example 3 except all actions are available to both agents, i.e., both P and Q can choose among actions $\{pA0, pA1, sA, pB0, pB1, sB\}$. We still assume that actions are private. As in example 2, we will show that there is not enough information to conclude the agents can open the safe. However, this is not because there is no good joint plan, instead the problem is that one agent might have extra knowledge which enables him to also open the safe on his own (if the other agent does not interfere) and since actions are private this can cause lack of coordination.

3 The formal framework

The basis of our framework for joint ability is the situation calculus [McCarthy and Hayes, 1969; Levesque *et al.*, 1998]. The situation calculus is a predicate calculus language for representing dynamically changing domains. A *situation* represents a possible state of the domain. There is a set of initial situations corresponding to the ways the domain might be initially. The actual initial state of the domain is represented by the distinguished initial situation constant, S_0. The term $do(a, s)$ denotes the unique situation that results from an agent doing action a in situation s. We use $do(\langle a_1, \cdots, a_n \rangle, s)$ as a shorthand for $do(a_n, do(\cdots, do(a1, s)) \cdots)$. Initial situations are defined as those that do not have a predecessor: $Init(s) \doteq \neg\exists a \exists s'.\ s = do(a, s')$. In general, the situations can be structured into a set of trees, where the root of each tree is an initial situation and the arcs are actions. The formula $s \sqsubseteq s'$ is used to state that there is a path from situation s to situation s'. Our account of joint ability will require some second-order features of the situation calculus, including quantifying over certain functions from situations to actions, that we call *strategies*.

Predicates and functions whose values may change from situation to situation (and whose last argument is a situation) are called *fluents*. The effects of actions on fluents are defined using successor state axioms [Reiter, 2001], which provide a succinct representation for both effect and frame axioms [McCarthy and Hayes, 1969]. To axiomatize a dynamic domain in the situation calculus, we use Reiter's [2001] action theory, which consists of (1) successor state axioms; (2) initial state axioms, describing the initial states of the domain including the initial beliefs of the agents; (3) precondition axioms, specifying the conditions under which each action can be executed; (4) unique names axioms for the actions, and (5) domain-independent foundational axioms (we adopt the ones given in [Levesque *et al.*, 1998] which accommodate multiple initial situations, but we do not describe them further here).

For our examples, we need eight fluents. The fluents *cA* and *cB* indicate the combination of locks A and B (i.e. true corresponds to button 1, and false corresponds to button 0 as the correct buttons that need to be pushed). The fluents *open*, *standby*, and *alarm* indicate whether the safe is open, the safe is on standby, and the alarm is activated, respectively. The fluent *time* indicates how many actions have been performed. Finally, the fluent *turn* is used to indicate whose turn it is to act, and the fluent B deals with the beliefs of the agents.

Moore [1985] defined a possible-worlds semantics for a logic of knowledge in the situation calculus by treating situations as possible worlds. Scherl and Levesque [2003] adapted this to Reiter's theory of action and gave a successor state axiom for B that states how actions, including sensing actions, affect knowledge. Shapiro *et al.* [1998] adapted this to handle the beliefs of multiple agents, and we adopt their account here. $B(x, s', s)$ will be used to denote that in situation s, agent x thinks that situation s' might be the actual situation. Note that the order of the situation arguments is reversed from the convention in modal logic for accessibility relations. Belief is then defined as an abbreviation:[1]

$$\mathit{Bel}(x, \phi[\mathit{now}], s) \doteq \forall s'.\ B(x, s', s) \supset \phi[s'].$$

We will also use the following abbreviation:

$$\mathit{BW}(x, \phi, s) \doteq \mathit{Bel}(x, \phi, s) \vee \mathit{Bel}(x, \neg\phi, s).$$

Mutual beliefs among the agents, denoted by *MBel*, can be defined either as a fix-point or by introducing a new accessibility relation using a second-order definition.

Our examples use the following successor state axioms:

- The combinations of the locks do not change over time: $\mathit{cA}(do(a, s)) \equiv \mathit{cA}(s)$, and $\mathit{cB}(do(a, s)) \equiv \mathit{cB}(s)$.
- If the safe is on standby and the alarm is not active, pushing the correct button for lock B opens the safe:
$\mathit{open}(do(a, s)) \equiv \mathit{standby}(s) \wedge \neg \mathit{alarm}(s) \wedge$
$[\mathit{cB}(s) \wedge a = \mathit{pB1} \vee \neg\mathit{cB}(s) \wedge a = \mathit{pB0}] \vee \mathit{open}(s)$.
- If the alarm is not active, pushing the correct button for lock A puts the safe on standby:
$\mathit{standby}(do(a, s)) \equiv \neg\mathit{alarm}(s) \wedge$
$[\mathit{cA}(s) \wedge a = \mathit{pA1} \vee \neg\mathit{cA}(s) \wedge a = \mathit{pA0}] \vee \mathit{standby}(s)$.
- The alarm is activated by pushing the wrong button, pushing a button of A if the safe is already on standby, or pushing any button if the safe is already open:
$\mathit{alarm}(do(a, s)) \equiv \mathit{alarm}(s)\ \vee$
$\mathit{cA}(s) \wedge a = \mathit{pA0}\ \vee\ \neg\mathit{cA}(s) \wedge a = \mathit{pA1}\ \vee$
$\mathit{cB}(s) \wedge a = \mathit{pB0}\ \vee\ \neg\mathit{cB}(s) \wedge a = \mathit{pB1}\ \vee$
$\mathit{standby}(s) \wedge (a = \mathit{pA0}\ \vee\ a = \mathit{pA1})\ \vee$
$\mathit{open}(s) \wedge [a = \mathit{pA0} \vee a = \mathit{pA1} \vee a = \mathit{pB0} \vee a = \mathit{pB1}]$.
- Belief changes due to sensing and other actions. We use the following type of successor state axiom proposed by Scherl and Levesque in the case where actions are public (see Section 4.3 for the case where actions are private):
$B(x, s', do(a, s)) \equiv \exists s''.\ B(x, s'', s) \wedge s' = do(a, s'')$
$\wedge\ [\mathit{agent}(a) = x \supset (\mathit{SF}(a, s'') \equiv \mathit{SF}(a, s))]$.
$\mathit{SF}(a, s) \equiv [a = \mathit{sA} \supset \mathit{cA}(s)] \wedge [a = \mathit{sB} \supset \mathit{cB}(s)]$.
Thus when any action occurs, all agents learn that it has occurred. Moreover, when an agent performs *sA* or *sB*, he alone learns the corresponding lock combination.
- Each action uses one time step:
$\mathit{time}(do(a, s)) = \mathit{time}(s) + 1$.
- Whose turn it is to act alternates between P and Q:
$\mathit{turn}(do(a, s)) = x \equiv$
$\mathit{turn}(s) = Q \supset x = P\ \wedge\ \mathit{turn}(s) = P \supset x = Q$.

The examples also include the following initial state axioms:

- $\mathit{Init}(s) \supset \mathit{turn}(s) = P$. So, agent P gets to act first.
- In all initial situations, time starts at 0, the alarm is not active, the safe is not on standby and not open:
$\mathit{Init}(s) \supset$
$\mathit{time}(s) = 0 \wedge \neg\mathit{alarm}(s) \wedge \neg\mathit{standby}(s) \wedge \neg\mathit{open}(s)$.
- Each agent initially knows that it is in an initial situation:
$\mathit{Init}(s) \wedge B(x, s', s) \supset \mathit{Init}(s')$.
- B models knowledge, and hence beliefs must be true:
$\mathit{Init}(s) \supset B(x, s, s)$.
- Each agent initially has introspection of her beliefs:
$\mathit{Init}(s) \wedge B(x, s', s) \supset [\forall s''.\ B(x, s'', s') \equiv B(x, s'', s)]$.

The last two properties of belief can be shown to hold for all situations using the successor state axiom for B so that belief satisfies the modal system *KT45* [Chellas, 1980]. Since the axioms above are universally quantified, they are known to all agents, and in fact are common knowledge. We will let Σ denote the action theory containing the successor and initial state axioms above. All the examples in Section 4 will use Σ (with variations in the B or *SF* axiom) and in some cases with additional conditions about the beliefs of agents.

3.1 Our definition of joint ability

In this paper, for simplicity, we use [Ghaderi *et al.*, 2007]'s definition restricted to two agents (for the general definition see [Ghaderi *et al.*, 2007]). All of the definitions below are abbreviations for formulas in the language of the situation calculus presented above. The joint ability of two agents P and Q to achieve ϕ is defined as follows:

- P and Q can jointly achieve ϕ starting from s iff all combinations of their preferred strategies work together:
$\mathit{JCan}(\phi, s) \doteq \forall \sigma_p, \sigma_q.\ \mathit{Pref}(P, \sigma_p, \phi, s) \wedge$
$\mathit{Pref}(Q, \sigma_q, \phi, s) \supset \mathit{Works}(\sigma_p, \sigma_q, \phi, s)$.
- The pair of strategies σ_p and σ_q works if there is a future situation where ϕ holds and the strategies prescribe the actions to get there according to whose turn it is:
$\mathit{Works}(\sigma_p, \sigma_q, \phi, s) \doteq$
$\exists s''.\ s \sqsubseteq s'' \wedge \phi[s''] \wedge \forall s'.\ s \sqsubseteq s' \sqsubset s'' \supset$
$(\mathit{turn}(s') = P \supset do(\sigma_p(s'), s') \sqsubseteq s'') \wedge$
$(\mathit{turn}(s') = Q \supset do(\sigma_q(s'), s') \sqsubseteq s'')$.
- Agent x prefers strategy σ_x if it is kept for all levels n:
$\mathit{Pref}(x, \sigma_x, \phi, s) \doteq \forall n.\ \mathit{Keep}(x, n, \sigma_x, \phi, s)$.
- *Keep* is defined inductively:[2]
  - At level 0, each agent keeps all of her strategies:
  $\mathit{Keep}(x, 0, \sigma_x, \phi, s) \doteq \mathit{Strategy}(x, \sigma_x)$.

[1] Free variables are assumed to be universally quantified from outside. If ϕ is a formula with a single free situation variable, $\phi[t]$ denotes ϕ with that variable replaced by situation term t. Instead of $\phi[\mathit{now}]$ we occasionally omit the situation argument completely.

[2] Strictly speaking, the definition we propose here is ill-formed. We want to use it with the second argument universally quantified (as in *Pref*). *Keep* and *GTE* actually need to be defined using second-order logic, from which the definitions here emerge as consequences. We omit the details for space reasons.

- at level $n+1$, agent x keeps strategy σ_x if it was kept at level n and there was *not* a *better* kept σ'_x (σ'_x is better than σ_x if σ'_x is as good as, i.e. greater than or equal to, σ_x while σ_x is not as good as it):
$\mathit{Keep}(x, n+1, \sigma_x, \phi, s) \doteq \mathit{Keep}(x, n, \sigma_x, \phi, s) \wedge \neg\exists\sigma'_x.\ \mathit{Keep}(x, n, \sigma'_x, \phi, s) \wedge \mathit{GTE}(x, n, \sigma'_x, \sigma_x, \phi, s) \wedge \neg\mathit{GTE}(x, n, \sigma_x, \sigma'_x, \phi, s)$.

- Strategy σ_x is as good as (Greater Than or Equal to) σ'_x for agent x at level n if x believes that whenever σ'_x works with strategies kept by the other agent y, so does σ_x. Note that here $x = P \wedge y = Q$ or $x = Q \wedge y = P$:
$\mathit{GTE}(x, n, \sigma_x, \sigma'_x, \phi, s) \doteq \forall\sigma_y.\ \mathit{Bel}(x, [\mathit{Keep}(y, n, \sigma_y, \phi, \mathit{now}) \wedge \mathit{Works}(\sigma'_x, \sigma_y, \phi, \mathit{now}) \supset \mathit{Works}(\sigma_x, \sigma_y, \phi, \mathit{now})], s)$.
- Finally, strategies for an agent are functions from situations to actions such that the required action is legal and known to the agent whenever it is the agent's turn to act:
$\mathit{Strategy}(x, \sigma) \doteq \forall s.\ \mathit{turn}(s) \neq x \supset \sigma(s) = \mathit{nil} \wedge \mathit{turn}(s) = x \supset \exists a.\, \mathit{Bel}(x, \sigma(\mathit{now}) = a, s) \wedge \mathit{Legal}(a)$.
Legal will depend on the domain. For examples 1, 2 and 3, it is defined such that P can only do actions *pA0*, *pA1*, and *sA*, while Q can only do *pB0*, *pB1*, and *sB*. For example 4, all actions will be possible for both agents.

These formulas define joint ability in a way that resembles the iterative elimination of weakly dominated strategies of game theory [Osborne and Rubinstein, 1999]. As we will see in the examples next, the mere *existence* of a working strategy profile is not enough; the definition requires coordination among the agents in that *all* preferred strategies must work together.

4 Formalizing the Examples

In this section, we prove results about the four examples mentioned earlier. Due to lack of space we present only brief proof sketches. Note that the goal in all examples is to open the safe in exactly 4 steps without activating the alarm, i.e.

$$\phi(s) \doteq \mathit{open}(s) \wedge \neg\mathit{alarm}(s) \wedge \mathit{time}(s) = 4.$$

4.1 Example 1

Recall that for this example actions are divided between agents and are public. We show that the agents are jointly able to achieve the goal (and have mutual belief about this):

Theorem 1 $\Sigma \models \mathit{Init}(s) \supset \mathit{JCan}(\phi, s)$.

Actually, it is sufficient to show that the following holds:

Theorem 2 $\Sigma \models \mathit{Init}(s) \wedge \mathit{Keep}(P, 2, \sigma_p, \phi, s) \wedge \mathit{Keep}(Q, 2, \sigma_q, \phi, s) \supset \mathit{Works}(\sigma_p, \sigma_q, \phi, s)$.

The proof is involved, so we just sketch the steps. Assume M is a model of Σ and μ a variable assignment such that $M, \mu \models \mathit{Init}(s) \wedge \mathit{Keep}(P, 2, \sigma_p, \phi, s) \wedge \mathit{Keep}(Q, 2, \sigma_q, \phi, s)$. We need to show that $\mathit{Works}(\sigma_p, \sigma_q, \phi, s)$ holds. Let $\overline{\sigma_p}$ and $\overline{\sigma_q}$ be strategies prescribing that P and Q initially sense the combination of locks A and B, respectively, and then push the correct button of the corresponding lock, in turn, i.e.:[3]

- $M, \mu \models \forall s, a1, a2, s'.\ \mathit{Init}(s) \supset \overline{\sigma_p}(s) = \mathit{sA} \wedge \overline{\sigma_p}(\mathit{do}(a1, s)) = \mathit{nil} \wedge \overline{\sigma_p}(\mathit{do}(\langle a1, a2\rangle, s)) = \mathit{pA}(s) \wedge [\mathit{do}(\langle a1, a2\rangle, s) \sqsubset s' \supset \mathit{turn}(s') = P \supset \overline{\sigma_p}(s') = \mathit{sA} \wedge \mathit{turn}(s') \neq P \supset \overline{\sigma_p}(s') = \mathit{nil}]$.
- $M, \mu \models \forall s, a1, a2, a3, s'.\ \mathit{Init}(s) \supset \overline{\sigma_q}(s) = \mathit{nil} \wedge \overline{\sigma_q}(\mathit{do}(a1, s)) = \mathit{sB} \wedge \overline{\sigma_q}(\mathit{do}(\langle a1, a2\rangle, s)) = \mathit{nil} \wedge \overline{\sigma_q}(\mathit{do}(\langle a1, a2, a3\rangle, s)) = \mathit{pB}(s) \wedge [\mathit{do}(\langle a1, a2, a3\rangle, s) \sqsubset s' \supset \mathit{turn}(s') = Q \supset \overline{\sigma_q}(s') = \mathit{sB} \wedge \mathit{turn}(s') \neq Q \supset \overline{\sigma_q}(s') = \mathit{nil}]$.

It can be easily shown that functions $\overline{\sigma_p}$ and $\overline{\sigma_q}$ are in fact strategies for P and Q that together achieve the goal in *all* initial situations, and hence each survives the first round of elimination for the corresponding agent. Also, after the first round of eliminations, the following holds *at level 1*:

Theorem 3 *$\overline{\sigma_p}$ and $\overline{\sigma_q}$ are as good as any other strategies:*

- $M, \mu \models \forall s, \sigma_p.\ \mathit{Init}(s) \wedge \mathit{Strategy}(P, \sigma_p) \supset \mathit{GTE}(P, 1, \overline{\sigma_p}, \sigma_p, \phi, s)$.
- $M, \mu \models \forall s, \sigma_q.\ \mathit{Init}(s) \wedge \mathit{Strategy}(Q, \sigma_q) \supset \mathit{GTE}(Q, 1, \overline{\sigma_q}, \sigma_q, \phi, s)$.

By theorem 3, $\mathit{GTE}(P, 1, \overline{\sigma_p}, \sigma_p, \phi, s)$ holds, and by assumption $\mathit{Keep}(P, 2, \sigma_p, \phi, s)$ holds, therefore we must have $\mathit{GTE}(P, 1, \sigma_p, \overline{\sigma_p}, \phi, s)$. Then, since in all initial situations $\mathit{Works}(\overline{\sigma_p}, \overline{\sigma_q}, \phi, s)$ holds, we must have $\mathit{Works}(\sigma_p, \overline{\sigma_q}, \phi, s)$. By a similar argument, we have $\mathit{GTE}(Q, 1, \sigma_q, \overline{\sigma_q}, \phi, s)$. This together with $\mathit{Works}(\sigma_p, \overline{\sigma_q}, \phi, s)$ obtained above, leads to $\mathit{Works}(\sigma_p, \sigma_q, \phi, s)$ as desired and thus Theorem 2 holds. ■

Theorem 3 itself can proved by the following two lemmas:

Lemma 1 *For any strategy for P that survives the first elimination round, if its first action is to push the correct A button, the action prescribed in response to Q doing sB must be sA:*
$\Sigma \models \mathit{Init}(s) \wedge \mathit{Keep}(P, 1, \sigma_p, \phi, s) \wedge \sigma_p(s) = \mathit{pA}(s) \supset \sigma_p(\mathit{do}(\langle \mathit{pA}(s), \mathit{sB}\rangle, s)) = \mathit{sA}$.

Lemma 2 *For any strategy for Q that survives the first round of elimination, its first action in response to P doing sA must be doing sB, and then if P continues by pushing the correct button of lock A, Q must push the correct button of B:*
$\Sigma \models \mathit{Init}(s) \wedge \mathit{Keep}(Q, 1, \sigma_q, \phi, s) \supset \sigma_q(\mathit{do}(\mathit{sA}, s) = \mathit{sB} \wedge \sigma_q(\mathit{do}(\langle \mathit{sA}, \mathit{sB}, \mathit{pA}(s)\rangle, s)) = \mathit{pB}(s)$.

The proofs of lemmas 1 and 2 are omitted but they use the fact that in a given initial situation s there are only 3 legal sequences of length 4 that can open the safe without activating the alarm: $[\mathit{sA}; \mathit{sB}; \mathit{pA}(s); \mathit{pB}(s)]$, $[\mathit{pA}(s); \mathit{sB}; \mathit{sA}; \mathit{pB}(s)]$, and $[\mathit{pA}(s); \mathit{pB}(s); \mathit{sA}; \mathit{sB}]$, where $\mathit{pA}(s)$ and $\mathit{pB}(s)$ correspond to the correct push actions for lock A and B in s, respectively.

We remind the reader that the reason that the above proofs are involved is that we have not specified anything about the beliefs of agents about the locks combinations and/or each other. Our theorems hold no matter what beliefs the agents have about this (e.g. if P and Q know the combination of both locks but neither knows what the other agent knows, they can still coordinate to open the safe despite the existence of many working plans). See example 4 as a case where the existence of multiple joint plans can cause lack of coordination.

[3] In what follows, we use $\mathit{pA}(s)$ as a shorthand for the *correct push action* for lock A in situation s. Any formula ψ that mentions $\mathit{pA}(s)$ with free variable s stands for $(\mathit{cA}(s) \supset \psi[\mathit{pA}(s)/\mathit{pA1}]) \wedge (\neg\mathit{cA}(s) \supset \psi[\mathit{pA}(s)/\mathit{pA0}])$, where $\psi[u/v]$ is replacing all free occurrences of u by v in ψ. We use a similar definition for $\mathit{pB}(s)$.

4.2 Example 2

In this example, action *sA* does not provide new information. To handle this, let Σ_2 be exactly like Σ except the *SF* axiom is replaced by $SF(a,s) \equiv [a = sB \supset cB(s)]$. The information in Σ_2 is not enough to conclude joint ability. In fact, we show that if P does not know the combination of lock A they *cannot* open the safe (even if Q knows both combinations):

Theorem 4 $\Sigma_2 \models \mathit{Init}(s) \land \neg \mathit{BW}(P, \mathit{cA}, s) \supset \neg \mathit{JCan}(\phi, s)$.

Proof sketch: Let M be a model of Σ_2 and μ be a variable assignment such that $M, \mu \models \mathit{Init}(s) \land \neg \mathit{BW}(P, \mathit{cA}, s)$, it is sufficient to show that there exists a pair of *preferred* strategies for P and Q that does not achieve the goal. Since P does not know the combination of A, there is at least another accessible initial situation s' such that $\mathit{cA}(s) \equiv \neg \mathit{cA}(s')$. Note that the function $\overline{\sigma_p}$ defined in example 1 is not a strategy in this model as P now does not know the combination of lock A even after performing *sA*. We can show that P has at least two preferred strategies σ_p and σ'_p such that $M, \mu \models \sigma_p(s) = \mathit{pA0} \ \land \ \sigma'_p(s) = \mathit{pA1}$. However, for *any* strategy σ_q for Q, one of the pair (σ_p, σ_q) or (σ'_p, σ_q) does not work in s, as the first action by P activates the alarm. ■

4.3 Example 3

In this example, everything is the same as in example 1 except that actions are now private, so the other agent does not see what actions are performed by the other agent (but each agent is aware of her own actions including the sensing results if any). To accommodate for this, we define Σ_3 exactly as Σ except we modify the successor state axiom for B as follows:

$$B(x, s', do(a, s)) \equiv \exists s'', a''.$$
$$B(x, s'', s) \land s' = do(a'', s'') \land \mathit{Legal}(a'') \land$$
$$[\mathit{agent}(a) = x \ \supset \ a = a'' \land (\mathit{SF}(a, s'') \equiv \mathit{SF}(a, s))].$$

Despite actions being private, we can prove that the agents have joint ability to open the safe (again without any need for additional specifications about their beliefs):

Theorem 5 $\Sigma_3 \models \mathit{Init}(s) \supset \mathit{JCan}(\phi, s)$.

The proof is similar to that of example 1. Note that $\overline{\sigma_p}$ and $\overline{\sigma_q}$ used there to eliminate non-promising strategies did not rely on actions of the other agent and are applicable here as well. However, the proof for Theorem 3 is slightly different.

4.4 Example 4

In this example, all actions are legal for both agents but, as in example 3, are private. To handle this, let Σ_4 be like Σ_3 except that *Legal* is defined such that both agents can perform any of actions *pA0*, *pB0*, *pA1*, *pB1*, *sA*, and *sB*.[4] Under these assumptions we cannot conclude that the agents have joint ability to open the safe; in fact we show that if it is initially mutually known that P does not know the lock combinations and Q knows both combinations they *cannot* open the safe:

Theorem 6 $\Sigma_4 \models \mathit{Init}(s) \land \mathit{MBel}(\mathit{BW}(Q, \mathit{cA}) \land \mathit{BW}(Q, \mathit{cB}) \land \neg \mathit{BW}(P, \mathit{cA}) \land \neg \mathit{BW}(P, \mathit{cB}), s) \supset \neg \mathit{JCan}(\phi, s)$.

[4]Technically, every action a has an agent parameter as its first argument where, e.g., $\mathit{agent}(\mathit{pA0}(x)) = x$. To simplify the presentation we have omitted the agent argument. Very minor modifications to the formulas presented here are needed to restore the argument.

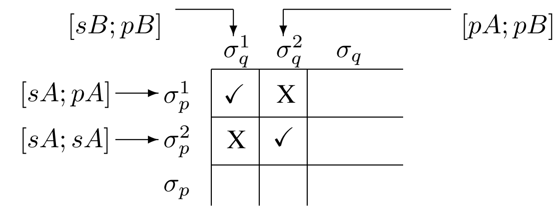

Figure 1: For private shared actions, if it is mutually believed that Q knows the locks combinations and P does not, multiple *incompatible preferred* plans exist that cause lack of coordination. In the above matrix, a ✓ at i, j corresponds to $\mathit{MBel}(\mathit{Works}(i, j, \phi), s)$ and an X corresponds to $\mathit{MBel}(\neg \mathit{Works}(i, j, \phi), s)$.

The interesting point here is that unlike in example 2, this is not because no joint plan exists. Quite the opposite, there are multiple joint plans that open the safe but the agents cannot coordinate (assuming no prior conventions can be relied upon). To see this consider any model M of Σ_4 and variable assignment μ where $M, \mu \models \mathit{Init}(s) \land \mathit{MBel}(\mathit{BW}(Q, \mathit{cA}) \land \mathit{BW}(Q, \mathit{cB}) \land \neg \mathit{BW}(P, \mathit{cA}) \land \neg \mathit{BW}(P, \mathit{cB}), s)$. We can show that P and Q each prefers at least two strategies whose combinations do not always work (i.e. there is lack of coordination). Let σ_p^1 be a strategy for P that prescribes sensing the combination of lock A and pushing its correct button as P's first and second (non-*nil*) actions (represented by [*sA*; *pA*]). Also, let σ_p^2 be a strategy for P that always prescribes performing *sA* whenever it is P's turn (represented by [*sA*; *sA*]). Similarly, let σ_q^1 be a strategy for Q that says to sense the combination of lock B and then to push its correct button as Q's first and second non-*nil* actions (represented by [*sB*; *pB*]). Finally, let σ_q^2 be a strategy for Q that prescribes pushing the correct button of lock A and B as Q's first and second non-*nil* actions (represented by [*pA*; *pB*]). Note that since Q knows both combinations, σ_q^2 is in fact a valid strategy. Clearly, we have $M, \mu \models \mathit{MBel}(\mathit{Works}(\sigma_p^1, \sigma_q^1, \phi) \land \mathit{Works}(\sigma_p^2, \sigma_q^2, \phi) \land \neg \mathit{Works}(\sigma_p^1, \sigma_q^2, \phi) \land \neg \mathit{Works}(\sigma_p^2, \sigma_q^1, \phi), s)$, see Fig. 1. To show that the agents are not able to open the safe, it remains to show that P and Q never eliminate these strategies:

Lemma 3 *P prefers σ_p^1 and σ_p^2. Q prefers σ_q^1 and σ_q^2:*

- $M, \mu \models \forall i.\ \mathit{Keep}(P, i, \sigma_p^1, \phi, s) \land \mathit{Keep}(P, i, \sigma_p^2, \phi, s)$.
- $M, \mu \models \forall i.\ \mathit{Keep}(Q, i, \sigma_q^1, \phi, s) \land \mathit{Keep}(Q, i, \sigma_q^2, \phi, s)$.

We sketch the proof for the 1st elimination round ($i = 1$), the generalization to all i's is done using simple induction. Assume to the contrary $M, \mu \models \neg \mathit{Keep}(P, 1, \sigma_p^1, \phi, s)$. Then there must exist a better strategy σ_p for P such that $\mathit{GTE}(P, 0, \sigma_p, \sigma_p^1, \phi, s)$ and $\neg \mathit{GTE}(P, 0, \sigma_p^1, \sigma_p, \phi, s)$. Hence, since $M, \mu \models \mathit{MBel}(\mathit{Works}(\sigma_p^1, \sigma_q^1, \phi), s)$, we must have $M, \mu \models \mathit{Bel}(P, \mathit{Works}(\sigma_p, \sigma_q^1, \phi), s)$. However, any strategy for P that works with σ_q^1 in *all* P's accessible initial situations must prescribe doing *sA* and then *pA* as P's first two non-*nil* actions, respectively.[5] Hence, the first two actions of σ_p and σ_p^1 are the same, which contradicts the assumption of σ_p being better than σ_p^1. Therefore, P keeps σ_p^1 at level 1. Similarly, we can show that if there were strategy σ_p better

[5]P does not know the combination of lock A, so there exist two accessible initial situations that differ on *cA*. Any strategy that prescribes first doing *pA0* (or *pA1*) activates the alarm in one of them.

than σ_p^2 then $M, \mu \models Bel(P, Works(\sigma_p, \sigma_q^2, \phi), s)$. However, any strategy σ_p that works with σ_q^2 in *all* P's accessible initial situations must prescribe doing nothing but sensing as P's first and second (non-*nil*) actions. It can then be shown that $M, \mu \models GTE(P, 0, \sigma_p^2, \sigma_p, \phi, s)$ which contradicts σ_p being better than σ_p^2. So, σ_p^2 is also kept at level 1. Finally, there are analogous arguments for Q keeping σ_q^1 and σ_q^2 at level 1. ■

5 Discussion and Future Work

In this paper, we extended Ghaderi *et al.* [2007]'s account of joint ability to domains with *sensing actions*, actions that allow agents to acquire new information as they proceed. We proposed ways of modeling the effects of such sensing actions on the agents' knowledge in the account. In such settings, strategies branch on sensing outcomes (as well as on observed actions by others), and the number of strategies typically grows extremely large. We showed that the symbolic proof techniques proposed in [Ghaderi *et al.*, 2007] could be generalized to establish joint ability or lack of joint ability in domains with sensing actions, even with very incomplete specifications of the agents' knowledge.

Our account of ability generalizes previous work on single agent ability [Moore, 1985; Davis, 1994; Lespérance *et al.*, 2000]. We go beyond these single agent accounts by modeling how the knowledge of all the agents changes as they act and by ensuring that the team remains coordinated — all of the agents' preferred strategies must work together.

Also related is work on logics of games [Pauly, 2002; van der Hoek and Wooldridge, 2003]. As mentioned earlier, these frameworks are propositional, and thus less expressive than ours. Moreover, they ignore the need for coordination inside a coalition, which is only reasonable if the agents can communicate arbitrarily to agree on a joint strategy.

Our approach goes beyond classical game theory [Osborne and Rubinstein, 1999] in that we can reason about joint ability even in the presence of incomplete specifications of the structure of the game including the beliefs of the agents. See [Ghaderi *et al.*, 2007] for more discussion of the relationship between the two accounts.

In this paper, for simplicity, we used Ghaderi *et al.*'s formalization of joint ability restricted to teams of two agents; see [Ghaderi *et al.*, 2007] for the general multiagent version. Their paper also discusses how agents that are outside of the team can be handled, i.e. by ensuring that the team's strategies achieve the goal for all of the outside agents' strategies.

As mentioned earlier, our approach can also handle *informing* communication actions where the truth value of a proposition or the value of a fluent is communicated by an agent to one or several other agents. It is straightforward to reformulate the examples considered in this paper to involve communication actions; instead of simply sensing a lock combination, an agent asks another "informer" agent for its value.

An issue for future work is examining how different ways of comparing strategies (the *GTE* order) affect the notion of joint ability. With the current *GTE* order, each agent compares her strategies by examining how they work when paired with the strategies of the other agent *in each accessible situation separately*. Another possibility is that, for example, each agent performs the comparison based on whether she *believes* her strategies work with those of the other agent (i.e. *Bel* is distributed over the implication in the *GTE* definition). Both definitions give the right results for our examples and others.

Also, in future work, we would like to generalize $Legal/Poss$ to be situation dependent, and devise ways of handling *conventions*, i.e. mutually believed rules that allow agents to stay coordinated. It would also be good to explore how the framework can be used in automated verification and in multiagent planning.

References

[Chellas, 1980] B. Chellas. *Modal logic: an introduction*. United Kingdom: Cambridge University Press, 1980.

[Davis, 1994] Ernest Davis. Knowledge preconditions for plans. *J. of Logic and Computation*, 4(5):721–766, 1994.

[Ghaderi *et al.*, 2007] Hojjat Ghaderi, Hector Levesque, and Yves Lespérance. A logical theory of coordination and joint ability. In *AAAI'07*, pages 421–426, 2007.

[Lespérance *et al.*, 2000] Yves Lespérance, Hector J. Levesque, Fangzhen Lin, and Richard B. Scherl. Ability and knowing how in the situation calculus. *Studia Logica*, 66(1):165–186, 2000.

[Levesque *et al.*, 1998] H. Levesque, F. Pirri, and R. Reiter. Foundations for the situation calculus. *Electronic Transactions on Artificial Intelligence*, 2(3-4):159–178, 1998.

[McCarthy and Hayes, 1969] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. 1969.

[Moore, 1985] R. Moore. A formal theory of knowledge and action. In J. Hobbs and R. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358, 1985.

[Osborne and Rubinstein, 1999] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1999.

[Pauly, 2002] M. Pauly. A modal logic for coalitional power in games. *Logic and Computation*, 12(1):149–166, 2002.

[Reiter, 2001] Raymond Reiter. *Knowledge in Action: Logical Foundations for Specifying & Implementing Dynamical Systems*. The MIT Press, 2001.

[Scherl and Levesque, 2003] R. Scherl and H. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144:1–39, 2003.

[Shapiro *et al.*, 1998] S. Shapiro, Y. Lespérance, and H. Levesque. Specifying communicative multi-agent systems. In W. Wobcke, M. Pagnucco, and C. Zhang, editors, *Agents and Multiagent systems*, pages 1–14. Springer-Verlag, 1998.

[van der Hoek and Wooldridge, 2003] W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

[Wooldridge and Jennings, 1999] M. Wooldridge and N. R. Jennings. The cooperative problem-solving process. *Journal of Logic and Computation*, 9(4):563–592, 1999.