Indexical Knowledge in Robot Plans*

Yves Lespérance and **Hector J. Levesque**†
Dept. of Computer Science
University of Toronto
Toronto, Ont. M5S 1A4 Canada
lesperan@ai.toronto.edu and hector@ai.toronto.edu

Abstract

Robots act upon and perceive the world from a particular perspective. It is important to recognize this relativity to perspective if one is not to be overly demanding in specifying what they need to know in order to be able to achieve goals through action. In this paper, we show how a formal theory of knowledge and action proposed in (Lespérance 1989) can be used to formalize several kinds of situations drawn from a robotics domain, where indexical knowledge is involved. Several examples treated deal with the fact that ability to act upon an object does not require *de re* knowledge of the object or its absolute position; knowledge of its relative position is sufficient. It is shown how the fact that perception yields indexical knowledge can be captured. We also point out the value of being able to relate indexical knowledge and objective knowledge within the same formalism through an example involving the use of a map for navigation. Finally, we discuss a problem raised by some higher-level parametrized actions and propose a solution.

Introduction

Robots act upon their world from a particular perspective, a particular place and moment in time. The same action done at different places and times has different effects; this is what we call the indexicality or context-sensitivity of action. Not surprisingly then, the knowledge that is directly required for action is often indexical, that is, relative to the agent's perspective. Similarly, the knowledge supplied by perception is indexical knowledge. Previous formal accounts of the ability of robots to achieve goals by doing actions, such as that of Moore (1980; 1985) and Morgenstern (1987), have ignored this, and thus end up imposing unnecessarily strong knowledge requirements upon agents before sanctioning their ability; they fail to properly specify the knowledge prerequisites and effects of actions. The deficiency is discussed by Haas (1986) within the context of a syntactic account of belief, but he does not formalize his proposals.

In a previous paper (Lespérance 1989), we proposed a theory of knowledge, action, and ability that captures the distinction between indexical knowledge and objective knowledge and permits a proper specification of the knowledge prerequisites and effects of actions. The functioning of the theory was then illustrated through the example of an agent making a phone call that may be long distance. In this paper, we examine applications of the theory in the robotics domain, where indexicality plays a particularly important role. We show how actions can be formalized, given that perception yields indexical knowledge, and that ability to act upon an object does not require *de re* knowledge of the object or its absolute position. We also show how indexical knowledge and objective knowledge can be related in our framework to deal with the use of maps for navigation. We discuss the representational issues that arise, which have general relevance to the formalization of actions with indexical knowledge prerequisites or effects. Our ability to handle these kinds of situation provides further evidence for the adequacy of our theory. Finally, we discuss problems that arise in handling some higher-level actions with object parameters. Before we can present these applications and discuss these issues, we must first introduce our framework; this is done in the next section.

The Formalism

Overview

Our theory of knowledge, action and ability is embodied in a first-order modal logic. The version described here involves several revisions from that presented in (Lespérance 1989); additional motivation is given there. A detailed exposition of the logic and its properties will be available in (Lespérance 1990).

The primary concern of our theory is the formalization of indexical knowledge, so let's start there. When one has indexical knowledge, for example when one knows that one is currently hungry, what is known is a "proposition" that is relative. It may be rela-

*This research was made possible in part by support from the Natural Science and Engineering Research Council of Canada.

†Fellow of the Canadian Institute for Advanced Research

tive to the knower, or to the time of the knowing, or perhaps to other aspects of the context. One typically expresses such relative propositions with sentences containing context-sensitive elements such as 'I', 'now', 'here', 'this person', *etc.* Our logic reflects this: its formulas contain elements whose interpretation depends on the context as well as the circumstances. The logical symbols **self** and **now** are indexical terms that refer respectively to the agent component and time component of the context.[1] Non-logical symbols will also typically depend on the time of the context for their evaluation; for example, HUNGRY(a) might represent the fact that the agent assigned to variable a is hungry at the time of the context. They can also depend on the agent component of the context; for example, THIRSTY might mean that the agent of the context is thirsty at the time of the context.

We model a context simply as a pair consisting of an agent and a time. This provides adequate semantics for many indexicals expressions. For instance, the indexical **here** can be taken to stand for the term POS(**self**), that is, 'the position of the agent of the context'. In other cases, one may be forced to be more specific than would be required in a natural language; for instance, one may have to say something like 'the person at such and such relative position from **self** at time **now**' rather than 'this person'. Informally, our logic only captures indexical expressions that are functions of **self** and **now**.[2]

Thus, our semantics evaluates formulas with respect to *indices*, which consist of a possible-world, modeling the objective circumstances, and an agent and time, modeling the context. We talk about a formula being satisfied by a model, *index*, and variable assignment.

Our semantics for knowledge is a simple generalization of the standard possible-world scheme (Kripke 1963): the knowledge accessibility relation K is taken to hold over indices. $\langle\langle \mathsf{w}, \mathsf{a}, \mathsf{t}\rangle, \langle \mathsf{w}', \mathsf{a}', \mathsf{t}'\rangle\rangle \in \mathsf{K}$ if as far as agent a at time t in world w knows, it may be the case that w' is the way the world actually is *and* he is a' *and* the current time is t'. Thus, we model the knowledge state of an agent at a time in a world by a set of indices, which characterizes not only which worlds are compatible with what the agent knows but also which points of views upon these worlds are compatible with what he knows.

[1] **self** and **now** are not intended to be formal counterparts of particular natural language words and often behave quite differently from any such counterparts.

[2] We view the question of whether the semantics of indexicals like 'you' and demonstratives like 'this person' can be captured by treating them as functions of **self** and **now** as open. Such a reduction would undoubtedly be complex, but it is not clear how such indexical expressions could play a causal role in cognition without there being such a reduction. But since we are primarily concerned with modeling action and its relationship to knowledge rather with providing a formal semantics for natural language, we can afford to remain uncommitted on this issue.

The formula **Know**(φ) is used to represent the fact that **self** (i.e., the agent of the context) knows at time **now** (i.e., the time of the context) that φ. If φ contains indexical elements, **Know**(φ) should be taken as attributing indexical knowledge, that is, knowledge the agent has about himself and the current time. For example, **Know**(HOLDING(x)) could mean that the agent knows that he himself is currently holding the object denoted by x.

An important advantage of our approach is that one can still model properties of knowledge by imposing constraints upon the accessibility relation K. We require K to be reflexive and transitive, which means that the principles of modal system S4 hold for knowledge. The approach can be adapted for modeling belief by changing the constraints. This treatment of indexical knowledge was inspired by informal proposals by Perry (1979) and especially Lewis (1979).

The operator **By** is used to say that an indexical "proposition" holds for an agent other than that of the context currently in effect. For example, **By**(a, **Know**(φ)) would say that agent a knows that φ. The argument formula is evaluated at an index whose agent component is the denotation of a.

We want to be able to make both eternal and indexical temporal assertions, express relations between indexically specified and objectively specified times, talk about agents knowing what time it is, etc. Due to this, time is reified in our logic, that is, terms that denote time points are included. Ordinary predicates, for instance HOLDING in the example above, are taken to represent static relations. As we have seen, atomic formulas involving such predicates are taken to assert that the associated relation holds at time **now**. One asserts that an agent does an action using the logical symbol **Does**, which functions as a predicate. For example, **Does**(GRASP, t) can be used to represent the fact that **self** does the action of grasping, from **now** to time t. The operator **At** is used to say that a state of affairs holds at a time other **now**. For example, **At**(t, HOLDING(x)) could mean that **self** is holding x at time t. The argument formula is evaluated at an index whose time is the denotation of t. The temporal aspects of the formalism were influenced by the work of Shoham (1987), Allen (1984), and McDermott (1982), as well as by the first-order temporal logic R described by Rescher and Urquhart (1971).

It is possible to express the occurrence of many types of complex actions using the constructs introduced above. We have developed a set of definitions that make it easy to state the occurrence of sequentially composed or conditional actions.[3]

Any account of the ability of agents to achieve goals

[3] Note that the expressive power of our temporal logic is not limited to this class of actions; actions involving nondeterminism, concurrency, multiple agents, definite times, etc. can be represented. But our formalization of ability is limited to actions belonging to this class.

by doing actions must be based on consideration of what actions are physically possible for agents and of what effects necessarily hold at the conclusion of these actions. In our framework, these notions are captured through modal operators for 'historical' necessity and possibility, whose semantics involves a relation over a set of possible courses of events (linear time frames). The operator $\Box$ corresponds to historical necessity, that is, what is necessary given everything that has happened up to and at time **now**. For example, $\Box(\mathbf{Does}(\textsc{pickup}, t) \supset \mathbf{At}(t, \textsc{Holding}(x)))$ says (rightly or wrongly) that if the agent does action PICKUP from **now** to time t, then he will necessarily be holding x at time t. Historical possibility ($\Diamond$) is defined in the usual way. This aspect of the framework is modeled on a system reviewed by Thomason (1984). The approach is more compatible with our account of knowledge, and the resulting system is more expressive than standard dynamic logic. To facilitate reasoning about action, we define the following notions:

$\mathbf{PhyPoss}(\delta) \stackrel{\text{def}}{=} \Diamond\exists v^t\, \mathbf{Does}(\delta, v^t)$, where v^t is a temporal variable that does not occur free in δ

$\mathbf{AfterNec}(\delta, \varphi) \stackrel{\text{def}}{=} \Box\forall v^t(\mathbf{Does}(\delta, v^t) \supset \mathbf{At}(v^t, \varphi))$, where v^t is a temporal variable that does not occur free in δ and φ

$\mathbf{Res}(\delta, \varphi) \stackrel{\text{def}}{=} \mathbf{PhyPoss}(\delta) \wedge \mathbf{AfterNec}(\delta, \varphi)$

$\mathbf{PhyPoss}(\delta)$ means that it is physically possible for **self** to do action δ next. $\mathbf{AfterNec}(\delta, \varphi)$ means that if **self** does δ next, it will necessarily be the case that φ holds afterwards. We say that δ results in φ, formally $\mathbf{Res}(\delta, \varphi)$, if δ is physically possible and after doing it, φ must hold.

Finally, our theory includes a formalization of ability which is a revised version of that of Moore (1980). First note that one may well know that cooking beef bourguignon would impress the party's guests without knowing how to do it, that is, without knowing what primitive actions "cooking beef bourguignon" really stands for. This also applies to actions that are instances of general procedures: if one ignores the combination of a safe, then one ignores what dialing the safe's combination amounts to (even though one knows the procedure for dialing any given combination). One can view this distinction as an instance of *de dicto* as opposed to *de re* knowledge. Moore exploits this in his formalization of ability. For simple actions, his account goes as follows: an agent a is able to achieve a goal by doing the action iff he knows what the given action is, knows that it is physically possible for a to do the action next, and knows that a's doing the action next necessarily results in the goal being achieved. By requiring that the agent know what the action is, Moore eliminates the need for explicit specification of the knowledge prerequisites of actions: if an action is an instance of a general procedure and the procedure is known (formally, an epistemically rigid function), then the action is known iff the agent knows what its arguments stand for. Complex actions are handled recursively. Note that it is not required that the agent initially know all the actions that make up a successful plan as long as he knows that he will know what to do next at each step of his plan.

The main deficiency we find in this formalization is that it requires the agent to know who he is (in an objective way). As we will argue below, this is neither necessary nor sufficient for ability. We will describe a revised version of the formalization which requires indexical knowledge instead of *de re* knowledge. Furthermore, the fact that our version is based upon a logic that includes an adequate treatment of indexical knowledge, allows actions with indexical knowledge prerequisites or effects to be properly formalized. We use the formula $\mathbf{Can}(\delta, \varphi)$ to express the fact that **self** is able to achieve the goal φ by doing action δ.

Semantics

A semantic structure M is a tuple

$$\langle \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{D}, \mathcal{W}, \prec, \mathsf{K}, \approx, \Phi, \Delta \rangle$$

The first four components are non-empty domains for the appropriate sorts: $\mathcal{A}$ is the domain of agents, $\mathcal{O}$ is the domain of objects, $\mathcal{T}$ is the domain of times, and $\mathcal{D}$ is the domain of primitive actions. The domain of individuals $\mathcal{I}$ is defined as $\mathcal{A} \cup \mathcal{O}$. $\mathcal{W}$ is a set of temporally extended possible worlds. $\mathcal{E} = \mathcal{W} \times \mathcal{A} \times \mathcal{T}$ is the set of indices. We take $\mathsf{a}, \mathsf{o}, \mathsf{i}, \mathsf{t}, \mathsf{d}, \mathsf{w}$, and e (possibly subscripted, primed, etc.), as ranging over arbitrary elements of $\mathcal{A}, \mathcal{O}, \mathcal{I}, \mathcal{T}, \mathcal{D}, \mathcal{W}$, and $\mathcal{E}$ respectively. $\prec$ is a strict total order on $\mathcal{T}$ whose intended interpretation is the relation "is earlier than". $\mathsf{K} \subseteq \mathcal{E}^2$ is the knowledge accessibility relation. The rationale behind this formulation was explained in the previous section. K must be reflexive and transitive.

The denotation of terms and satisfaction of formulas are defined relative to indices. Φ gives the extension of predicate and function symbols at an index. $\Delta \subseteq \mathcal{D} \times \mathcal{E} \times \mathcal{T}$ determines which actions are done by which agents in which worlds over which time intervals: $\langle \mathsf{d}, \langle \mathsf{w}, \mathsf{a}, \mathsf{t}_s \rangle, \mathsf{t}_e \rangle \in \Delta$ if action d is done by agent a from time t_s to time t_e in world w.

$\approx$ is a family of accessibility relations — one for each time point — that is used to interpret the historical necessity operator $\Box$. Intuitively, $\mathsf{w} \approx_{\mathsf{t}} \mathsf{w}^*$ if w and w* differ only in what happens after t. We ensure that our semantics respects this intended interpretation by imposing various constraints on $\approx$. Firstly, for all $\mathsf{t} \in \mathcal{T}$, $\approx_{\mathsf{t}}$ must be an equivalence relation — this implies that at any given time point, $\Box$ and $\Diamond$ obey the principles of the modal system S5. Secondly, if $\mathsf{w} \approx_{\mathsf{t}_2} \mathsf{w}^*$ and $\mathsf{t}_1 \preceq \mathsf{t}_2$, then $\mathsf{w} \approx_{\mathsf{t}_1} \mathsf{w}^*$, *i.e.* possibilities do not increase as time passes. And finally, to ensure that historical alternatives up to a given time are identical in what facts hold, what is done, and what is known up to that time, we require that if $\mathsf{w} \approx_{\mathsf{t}_2} \mathsf{w}^*$ and $\mathsf{t}_1 \preceq \mathsf{t}_2$, then

1. for any predicate $\boldsymbol{R}$,
$\Phi(\boldsymbol{R}, \langle w^*, a, t_1\rangle) = \Phi(\boldsymbol{R}, \langle w, a, t_1\rangle)$,
2. for any function symbol $\boldsymbol{f}$,
$\Phi(\boldsymbol{f}, \langle w^*, a, t_1\rangle)(i_1, \ldots, i_n) = \Phi(\boldsymbol{f}, \langle w, a, t_1\rangle)(i_1, \ldots, i_n)$,
3. $\Delta(d, \langle w^*, a, t_1\rangle, t_e)$ iff $\Delta(d, \langle w, a, t_1\rangle, t_e)$,
4. $\langle\langle w^*, a, t_1\rangle, e\rangle \in K$ iff $\langle\langle w, a, t_1\rangle, e\rangle \in K$.

To simplify reasoning about agents' ability to achieve goals by doing multi-step actions, we will assume that knowledge is persistent, that is, that agents keep on knowing what they knew previously, and that agents know what actions they have done. The following assumption enforces this: if $\langle\langle w, a, t_2\rangle, \langle w', a', t_2'\rangle\rangle \in K$ and $t_1 \preceq t_2$, then there exists a time $\hat{t}_1$, where $\hat{t}_1 \preceq t_2'$, such that $\langle\langle w, a, t_1\rangle, \langle w', a', \hat{t}_1\rangle\rangle \in K$ and if $\Delta(d, \langle w, a, t_1\rangle, t_2)$ then $\Delta(d, \langle w', a', \hat{t}_1\rangle, t_2')$. Note this formulation does not require agents to know at what time they start or finish acting, or how much time the action takes.

An assignment is a function that maps variables into elements of the domain appropriate to them. The *denotation* of a term θ in a structure M at an index $e = \langle w, a, t\rangle$ under an assignment g, written $[\![\theta]\!]^{M}_{e,g}$ is defined in the standard way for variables and compound terms; for indexicals, we have that $[\![\mathbf{self}]\!]_{e,g} = a$ and $[\![\mathbf{now}]\!]_{e,g} = t$ (when the structure under consideration is clear from context, we omit it). We can now define what it means for a formula φ to be *satisfied* by a structure M, an index $e = \langle w, a, t\rangle$, and an assignment g, which we write $M, e, g \models \varphi$. For conciseness, we omit the standard part of the definition that deals with with first-order logic with equality; for the rest of the language, we have:

$e, g \models \mathbf{Does}(d, t)$ iff $\Delta([\![d]\!]_{e,g}, e, [\![t]\!]_{e,g})$

$e, g \models t_1 < t_2$ iff $[\![t_1]\!]_{e,g} \prec [\![t_2]\!]_{e,g}$

$e, g \models \mathbf{At}(t, \varphi)$ iff $\langle w, a, [\![t]\!]_{e,g}\rangle, g \models \varphi$

$e, g \models \mathbf{By}(a, \varphi)$ iff $\langle w, [\![a]\!]_{e,g}, t\rangle, g \models \varphi$

$e, g \models \mathbf{Know}(\varphi)$ iff for all e', such that $\langle e, e'\rangle \in K$, $e', g \models \varphi$

$e, g \models \Box\varphi$ iff for all w^* such that $w \approx_t w^*$, $\langle w^*, a, t\rangle, g \models \varphi$

A formula φ is *satisfiable* iff there exists a structure M, index e, and assignment g, such that $M, e, g \models \varphi$. A formula φ is *valid* (written $\models \varphi$) iff it is satisfied by all structures, indices, and assignments.

Ability

Our current formalization of ability is based on that of Moore (1980). It is simpler than his because we do not attempt to handle indefinite iteration (while-loop actions). Moore's formalization of this case is actually defective because it does not require the agent to know that the action will eventually terminate. We leave this case for future research.

Since we are not treating indefinite iteration, we can simply define ability in terms of the other constructs of the logic as follows:

$\mathbf{Can}(\theta^d, \varphi) \stackrel{\text{def}}{=} \exists v^d\, \mathbf{Know}(v^d = \theta^d \wedge \mathbf{Res}(\theta^d, \varphi))$ where θ^d is an action term and action variable v^d does not occur free in φ and θ^d

$\mathbf{Can}(\mathbf{skip}, \varphi) \stackrel{\text{def}}{=} \mathbf{Know}(\varphi)$

$\mathbf{Can}((\delta_1; \delta_2), \varphi) \stackrel{\text{def}}{=} \mathbf{Can}(\delta_1, \mathbf{Can}(\delta_2, \varphi))$

$\mathbf{Can}(\mathbf{if}(\varphi_?, \delta_1, \delta_2), \varphi_g) \stackrel{\text{def}}{=} (\mathbf{Know}(\varphi_?) \wedge \mathbf{Can}(\delta_1, \varphi_g)) \vee (\mathbf{Know}(\neg\varphi_?) \wedge \mathbf{Can}(\delta_2, \varphi_g))$

The definition works by recursion on the structure of the action expressions involved. The first case handles simple actions (action terms): **self** is able to achieve a goal φ by doing a simple action θ^d iff he knows what that action is and knows that his doing it results in the goal holding afterwards. Note that the definition involves quantifying-in only over the class of primitive actions, which are agent-relative entities (e.g. "send grasping signal to hand"), quite unlike people or blocks. The second case states that **self** can achieve a goal by doing the empty action **skip** iff he knows that the goal currently holds. The third case says that **self** is able to achieve a goal φ by doing a sequentially composed action $(\delta_1; \delta_2)$ iff by doing δ_1, he is able to achieve the goal that consists in himself being able to achieve the original goal φ by doing δ_2. The final case takes care of conditional actions: **self** can achieve a goal by doing $\mathbf{if}(\varphi_?, \delta_1, \delta_2)$ iff he either knows that the condition $\varphi_?$ holds and is able to achieve the goal by doing δ_1, or knows that it does not hold and is able to achieve the goal by doing δ_2.

Our formalization improves over Moore's in several ways. Firstly, the simple action case requires the agent to know that if *he himself* does the action, the goal will necessarily hold afterwards; requiring the agent to know *of himself (de re)* that if he does the action the goal will necessarily hold afterwards, as Moore does, is neither necessary nor sufficient for the agent to be able to achieve the goal. We illustrate this through an example in the next section. Secondly, it is based on a very expressive temporal logic and thus could be more easily extended to handle actions that refer to times in more general ways than are considered here (*e.g.* the action of "running until noon"). Finally, the underlying logic includes an adequate treatment of indexical knowledge in general, which permits a more accurate specification of the knowledge prerequisites and effects of actions; the examples in the next section are evidence for this.

Formalizing a simple robotics domain

We will now use the theory to formalize aspects of a simple robotics domain and show how the resulting formalization can be used to prove various statements concerning the ability of agents to achieve goals by doing actions. We will argue that our framework allows a much more accurate modeling of these situations than frameworks that ignore indexicality. Our domain involves a robot, call him ROB, that moves about on a two-dimensional grid. Since our purpose is not to model complex patterns of interactions, but to present and justify our account of indexical knowledge and action, our formalization will be based on the assumption that the robot is the only source of activity in the domain. We take our robot to have the following repertory of basic actions (primitives of his architecture): he may move forward by one square, he may turn right or left 90°, he may sense whether an object is on the square where he is currently positioned and if there is one, what shape it has, and he may pick up an object from the current square or put down the object he is holding on the current square. It should be clear that in spite of the simplicity of this domain, it contains analogues to a large number of problems encountered in planning actual robot navigation, manipulation, and perception. For instance, one can view objects of particular shapes as landmarks and the robot can then navigate by recognizing such landmarks. We assume that there are no physical obstacles to the robot's movements; in particular, an object being on a square does not prevent the robot from being on it too (one can imagine the robot as standing over the object).

The indexicality of action manifests itself in many ways in this domain. One key way is that a robot can act upon (manipulate) an object as long as he knows where that object is *relative to himself*; he need not know either the object's absolute position or his own. First consider a simple instance of this where the robot wants to pick up an object and is actually positioned where that object is. Relevant aspects of the domain are formalized by making various assumptions, most of which have to do with the types of action involved. The following assumption specifies the effects of the action PICKUP:

Assumption 1 (Effects of PICKUP)

$$\models \forall x(\textsc{Object}(x) \wedge \textsc{pos}(x) = \mathbf{here} \wedge \neg\exists y\, \textsc{Holding}(y) \supset \mathbf{Res}(\textsc{pickup}, \textsc{Holding}(x)))$$

It says that if some object x is positioned where the agent currently is and he is not currently holding anything, then his doing the action PICKUP next will result in his holding x.[4] This means that under these conditions, it is both physically possible for him to do PICKUP, and his doing so necessarily results in his holding the object. In fact we assume that all basic actions are always possible. The view adopted is that such actions characterize essentially internal events which may have various external effects depending on the circumstances.[5] We also assume that agents always know how to do basic actions, that is, know what primitive actions they denote. This is formalized as follows:

Assumption 2 (Basic actions are rigid)

$\models \exists d\mathbf{Know}(d = \theta)$, where θ is any basic action constant

We omit the frame "axioms" for PICKUP, which say that it does not affect the position or orientation of anything and that unheld objects that are not where the agent is remain unheld.

Now clearly, just having *de re* knowledge of some object (i.e., $\exists x\, \mathbf{Know}(\textsc{Object}(x))$) is insufficient for being able to pick it up; something must be known about the object's position. If we only wanted to model the agent at a high level of abstraction, we might be willing to assume that as soon as an agent knows which object is involved, he would know how to get to it (or how to find out). But there clearly are circumstances where such assumptions are invalid and modeling at such an abstract level would leave out a great deal about how action is actually produced. We want an account that addresses the issue of what information the agent must exploit in order to be able to get at the object.

In a discussion of the robot action of "putting a block on another block", Moore (1985) recognizes this and suggests that it be defined in terms of lower-level actions involving arm motions to the objects' positions, grasping, and ungrasping. But, knowledge of an object's absolute position is not sufficient for being able to act upon it. One may not know what one's absolute position and orientation is and therefore may not be able to deduce where the object is relative to oneself. Our formalization reflects this fact: one can prove the following proposition with respect to the simple situation discussed earlier:

Proposition 1

$$\not\models \exists \vec{p}(\mathbf{here} = \vec{p} \wedge \mathbf{Know}(\exists x(\textsc{Object}(x) \wedge \textsc{pos}(x) = \vec{p}) \wedge \neg\exists y\, \textsc{Holding}(y))) \supset \mathbf{Can}(\textsc{pickup}, \exists x\, \textsc{Holding}(x)))$$

This says that even if the agent is currently at some position $\vec{p}$ and knows that the absolute position of some object is $\vec{p}$ and that he is not holding anything, he still might not be able to achieve the goal of holding some object by doing the action PICKUP. The reason for this is simply that the agent may not know that he is at $\vec{p}$.

On the other hand, we can also prove that if the agent knows that some object is *where he currently is* and that he is not holding anything, then he must be able to achieve the goal of holding some object by doing PICKUP:

Proposition 2

$$\models \mathbf{Know}(\exists x(\textsc{Object}(x) \wedge \textsc{pos}(x) = \mathbf{here}) \wedge \neg\exists y\, \textsc{Holding}(y)) \supset \mathbf{Can}(\textsc{pickup}, \exists x\, \textsc{Holding}(x))$$

[4]We assume that this holds for all agents rather than specifically about our robot because we want to avoid assuming that he knows who he is when we later prove statements about his abilities.

[5]Note that assumption 1 only specifies what happens when PICKUP is done under the conditions stated. What its effects are in other circumstances is not addressed.

The agent can be totally ignorant of what his (and the object's) absolute position is and still be able to achieve the goal.

Note that proposition 2 makes no requirement that the object that the agent ends up holding be the same as the one that was at his position before the action. This may appear too weak and an easy fix would involve assuming that the agent knows which object is involved. But is possible to strengthen the above proposition without requiring such *de re* knowledge. For example, the following proposition captures the fact that the agent knows that after the action, he would be holding some object that was where he was before doing the action.

Proposition 3

$$\models \mathbf{Know}(\exists x\varphi \wedge \neg\exists y\ \text{HOLDING}(y)) \supset \mathbf{Can}(\text{PICKUP}, \exists x(\exists t(t = \mathbf{now} \wedge \mathbf{Past}(\mathbf{Does}(\text{PICKUP}, t) \wedge \varphi)) \wedge \text{HOLDING}(x)))$$

where $\varphi \stackrel{\text{def}}{=} \text{OBJECT}(x) \wedge \text{POS}(x) = \mathbf{here}$

Specifically, it says that if the agent knows that some object is currently at his position and that he is not currently holding anything, then he can by doing action PICKUP achieve the goal of holding some object that was at his own position before the PICKUP he has just done ($\mathbf{Past}(\varphi)$ means that φ holds at some time earlier than **now**). This can be strengthened further to require uniqueness. But it should be clear that identifying the objects involved in the initial and goal situation, without requiring that it be known what objects they are, is not a trivial matter.

Before moving on, let's examine another variant of this situation. First, imagine that an agent a knows that there is an object where he himself is and that he is not holding anything. Then a is able to achieve the goal of holding something by doing PICKUP. Formally, if we let φ be the formula of proposition 2, then $\models \mathbf{By}(a, \varphi)$. However, if we imagine that a instead knows that there is an object where a is, it no longer follows that he is able to achieve the goal. That is, if φ' is the result of replacing **here** by $\text{POS}(a)$ in φ, we have that $\not\models \mathbf{By}(a, \varphi')$. The reason why this is not valid is simply that a may not know that he is a. This shows that knowing of oneself (*de re*) that if one does the action, the goal will necessarily hold afterwards, as Moore's formalization of ability requires, is not sufficient for ability. One can similarly show that such *de re* knowledge is not necessary either (in some models of proposition 2, the agent does not have such knowledge).

More generally, knowing the relative position of an object is sufficient for being able to act upon it. For instance, if the robot knows that there is an object at position $\langle 1, 0\rangle$ relative to himself, that is, on the square directly in front of him, and knows that he is not holding anything, then he is able to achieve the goal of holding some object by doing first FORWARD and then PICKUP. This can be proven given the assumption already stated and the following formalization of the action FORWARD:

Assumption 3 (Effects of FORWARD)

$$\models \forall\vec{p}\forall o(\mathbf{here} = \vec{p} \wedge \text{ORI} = o \supset \mathbf{Res}(\text{FORWARD}, \mathbf{here} = \vec{p} + \langle 1, 0\rangle \times \text{ROT}(o)))$$

Definition 1 $\text{ROT}(o) \stackrel{\text{def}}{=} \begin{pmatrix} \cos o & \sin o \\ -\sin o & \cos o \end{pmatrix}$

Assumption 4 $\models \forall x(\text{HOLDING}(x) \supset \text{POS}(x) = \mathbf{here})$

Assumption 3 says that as a result of doing FORWARD, the agent moves one square further along the direction he is facing; ORI represents the orientation of the agent with respect to the absolute frame of reference and $\text{ROT}(o)$ is the rotation matrix associated with angle o. Assumption 4 says that objects held by the agent are where he is. We also need the following three frame "axioms": firstly, after doing FORWARD, the agent's orientation must remain unchanged; secondly, after the agent does FORWARD, the position of objects that are not held by the agent must remain the same as before; and finally, objects that are not held by the agent must remain unheld after he does FORWARD.

Given this, we can prove proposition 4, that is, that if the agent knows that there is an object at position $\langle 1, 0\rangle$ relative to himself and that he is not holding anything, then he can achieve the goal of holding some object by doing FORWARD and then PICKUP:

Proposition 4

$$\models \mathbf{Know}(\exists x(\text{OBJECT}(x) \wedge \text{RPOS}(x) = \langle 1, 0\rangle) \wedge \neg\exists y\ \text{HOLDING}(y)) \supset \mathbf{Can}((\text{FORWARD}; \text{PICKUP}), \exists x\ \text{HOLDING}(x))$$

Definition 2

$$\text{RPOS}(x) \stackrel{\text{def}}{=} ((\text{POS}(x) - \mathbf{here}) \times \text{ROT}(-\text{ORI}))$$

$\text{RPOS}(x)$ represents the position of x relative to **self**. It is possible to prove the more general result that if the agent knows the relative position of an object and is not holding anything, he can go to that object's position and pick it up (since there are no obstacles, a trivial algorithm will achieve this).

We will come back to this issue of what one must know in order to be able to go and manipulate an object, but now let's have a look at perception. As observed earlier, it too yields indexical knowledge. In our domain, the action SENSE constitutes a limited form of perception. We formalize the effects of SENSE as follows:

Assumption 5

$$\models \mathbf{Res}(\text{SENSE}, \mathbf{Kwhether}(\exists x(\text{OBJECT}(x) \wedge \text{POS}(x) = \mathbf{here})))$$

Assumption 6

$$\models \forall s(\varphi \wedge \neg\exists y \text{HOLDING}(y) \supset \mathbf{Res}(\text{SENSE}, \mathbf{Know}(\varphi)))$$

where $\varphi \stackrel{\text{def}}{=} \exists x(\text{OBJECT}(x) \wedge \text{POS}(x) = \mathbf{here} \wedge \text{OFSHAPE}(x, s))$

Assumption 5 says that doing SENSE, results in the agent knowing whether[6] an object is present at his current position. Assumption 6 says that if some object is present at the agent's position and the agent is not holding anything, then his doing SENSE results in him knowing which shape(s) the object has. From this and the fact that basic actions are assumed to be known, it follows trivially that by doing SENSE, the agent can find out if there is an object where he is and, if there is one, what it's shape(s) is (are).

We exploit our formalization of SENSE in the following example drawn from another interesting area of robotics: that of navigation with the help of a map. In order to fully take advantage of the information contained in a map, say to find out how to get to a destination, an agent must first orient himself with respect to it, that is, find out where he is on the map, what his absolute position is. If he does not already know this, he must try to match the landmarks represented on the map with features of his current environment. Our simple domain provides instances of this if we treat objects of various shapes as landmarks. For example, one can prove the following proposition, which says that if an agent happens to be at absolute position $\vec{p}$ and knows that the unique object having shape s is at position $\vec{p}$, then he can find out that his absolute position is $\vec{p}$ by doing the action SENSE:

Proposition 5

$$\begin{aligned}\models \forall\vec{p}\forall s(&\mathbf{here} = \vec{p} \wedge \\ &\mathbf{Know}(\exists x(\textsc{Object}(x) \wedge \textsc{pos}(x) = \vec{p} \wedge \textsc{USh}(x,s)) \\ &\quad \wedge \neg\exists y\, \textsc{Holding}(y)) \\ &\supset \mathbf{Can}(\textsc{sense}, \mathbf{Know}(\mathbf{here} = \vec{p})))\end{aligned}$$

where $\textsc{USh}(x,s) \stackrel{\text{def}}{=} \forall y(\textsc{OfShape}(y,s) \equiv y = x)$

We can also show that an agent can find out what his absolute orientation is by recognizing objects that have a known orientation with respect to each other. And we can show that once an agent knows his absolute position, he can use the map to navigate to some object represented on it (see (Lespérance 1990) for details). These map navigation examples exploit a key feature of our framework, which is that it allows both indexical and absolute knowledge to be represented, and relations between the two to be expressed (this feature of the map navigation problem was pointed out by Israel (1987)). This distinguishes it from the indexical version of the situation calculus proposed by Subramanian and Woodfill (1989), where one simply introduces indexical entities in the ontology.

Let's now go back to the issue of what one must know in order to be able to go and act upon an object. We said that knowing the relative position of the object was sufficient for this. But in real life, agents rarely know exactly what the relative locations of objects are. More typically, they know roughly where objects are and scan the general area until they find the object. For instance, if our robot knows that an object is either on the square where he is or on the one directly in front of him, then he can achieve holding some object by first doing SENSE, and then either PICKUP, or FORWARD followed by PICKUP, according to whether the object turned out to be where he was or not. Another more complex instance goes as follows: if our robot knows that there is an object that is positioned at most k squares along the row he is facing (and that he is not holding anything), then he can get to it by repetitively moving forward and sensing (up to k times) until he senses that an object is present (Lespérance 1990).

So it is quite clear that ability to act upon an object does not require knowing its relative position. But then what is required? It seems that the best we can say is that the agent must *know of some procedure* that will take him to where the object is.

But this creates problems in the formalization of ability to do certain high-level parametrized actions, for example, the action of "going to the position of an object θ" GOWHERE(θ). It would be inappropriate to treat this action as a primitive because we want we want to model how knowledge enables action at a more detailed level. The other way to way to deal with such an action within our (and Moore's) framework would involve defining it in terms of lower-level actions that are parametrized with the information that must actually be known in order to be able to do the high-level action (recall Moore's proposal for the action of "putting a block on another"). This allows knowledge prerequisites to be enforced by the requirement that one know which primitive action to do next and removes the need to formalize them explicitly. But for actions like GOWHERE(θ), it is not clear how this could be put into practice.

However, notice that GOWHERE(θ) is a strange kind of action in that it appears to refer to anything that would achieve the goal that the agent be where θ is; it is as much like a *goal* as like an action. Perhaps we should rule out the introduction of such actions, but instead provide an action-less version of the **Can** operator: **CanAch**(φ) would mean that **self** is able to achieve φ in one way or another. Then, we may use **CanAch**(POS(θ) = **here** $\wedge$ φ) instead of something like **Can**(GOWHERE(θ), φ).[7] A coarse "syntactic" way of formalizing **CanAch** goes as follows: $\mathbf{e},\mathbf{g} \models \mathbf{CanAch}(\varphi)$ iff there exists an action expression δ such that $\mathbf{e},\mathbf{g} \models \mathbf{Can}(\delta,\varphi)$. A more general and robust approach is being developed by Nunes (1990).

Conclusion

Robots act upon and perceive the world from a particular perspective. It is important to recognize this

[6]**Kwhether**(φ) is defined as **Know**(φ) $\vee$ **Know**($\neg\varphi$).

[7]This assumes that it is known that θ refers to the same entity before and after the action is done; the assumption can be dispensed with by referring to the denotation of θ prior to the action as illustrated earlier.

relativity to perspective if one is not to be overly demanding in specifying what they need to know in order to be able to achieve goals through action. In a previous paper (Lespérance 1989), we proposed a formal theory of knowledge and action that accommodates the necessary indexical knowledge, and showed how it could be used to analyze an agent's ability to make a phone call that might be long-distance. Here, we have shown how the same framework can be used to formalize several kinds of situations involving indexicality drawn from a robotics domain. Several examples treated dealt with the fact that ability to act upon an object does not require *de re* knowledge of the object or its absolute position; knowledge of its relative position is sufficient. It was shown how the fact that perception yields indexical knowledge can be captured. We also pointed out the value of being able to relate indexical knowledge and objective knowledge within the same formalism through an example involving the use of a map for navigation. Finally, we discussed problems that arise in handling certain higher-level parametrized actions and proposed a solution.

We are examining further the role played by *de re* knowledge in action, with a view towards clarifying the notion and ensuring that our formalization of this role is adequate. Also under investigation are extensions to the theory to handle actions involving indefinite iteration. We are also applying the theory to a domain involving search through a data structure, to show that indexicality is not restricted to domains involving physical space. Applications to problems in models of linguistic communication are also under development.

Acknowledgements

Graeme Hirst, Joe Nunes, Adam Grove, Joe Halpern, Bart Selman, Gerhard Lakemeyer, and others have provided much useful advice and discussion.

References

Allen, J. F. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

Haas, A. R. 1986. A syntactic theory of belief and action. *Artificial Intelligence*, 28:245–292.

Israel, D. J. 1987. The role of propositional objects of belief in action. Technical Report CSLI-87-72, CSLI, Stanford University, Stanford, CA.

Kripke, S. A. 1963. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94.

Lespérance, Y. 1989. A formal account of self-knowledge and action. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 868–874, Detroit. Morgan Kaufman Publishing.

Lespérance, Y. 1990. *A Formal Theory of Indexical Knowledge and Action.* PhD thesis, Department of Computer Science, University of Toronto. To appear.

Lewis, D. 1979. Attitudes *de dicto* and *de se*. *The Philosophical Review*, 88(4):513–543.

McDermott, D. V. 1982. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155.

Moore, R. C. 1980. Reasoning about knowledge and action. Technical Report 191, AI Center, SRI International, Menlo Park, CA.

Moore, R. C. 1985. A formal theory of knowledge and action. In Hobbs, J. R. and Moore, R. C., editors, *Formal Theories of the Common-sense World.* Ablex Publishing, Norwood, NJ.

Morgenstern, L. 1987. Knowledge preconditions for actions and plans. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 867–874, Milan. Morgan Kaufman Publishing.

Nunes, J. H. T. 1990. Ability and commitment. Unpublished manuscript, Department of Computer Science, University of Toronto.

Perry, J. 1979. The problem of the essential indexical. *Noûs*, 13:3–21.

Rescher, N. and Urquhart, A. 1971. *Temporal Logic.* Springer-Verlag, Vienna.

Shoham, Y. 1987. Temporal logics in AI: Semantical and ontological considerations. *Artificial Intelligence*, 33(1):89–104.

Subramanian, D. and Woodfill, J. 1989. Making the situation calculus indexical. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 467–474, Toronto, Canada. Morgan Kaufman Publishing.

Thomason, R. C. 1984. Combinations of tense and modality. In Gabbay, D. and Guenthner, F., editors, *Handbook of Philosophical Logic*, volume 2, pages 135–165. D. Reidel Publishing, Dordrecht, Holland.