Robotics and the Common Sense Informatic Situation

Murray Shanahan[1]

Abstract. This paper proposes a logic-based framework in which a robot constructs a model of the world through an abductive process whereby sensor data is explained by hypothesising the existence, locations, and shapes of objects. Symbols appearing in the resulting explanations acquire meaning through the theory, and yet are grounded by the robot's interaction with the world. The proposed framework draws on existing logic-based formalisms for representing action, continuous change, space, and shape.

INTRODUCTION

Without ignoring the lessons of the past, the nascent area of Cognitive Robotics [Lespérance, *et al.*, 1994] seeks to reinstate the ideals of the Shakey project, namely the construction of robots whose architecture is based on the idea of representing the world by sentences of formal logic and reasoning about it by manipulating those sentences. The chief benefits of this approach are,

- that it facilitates the endowment of a robot with the capacity to perform high-level reasoning tasks, such as planning, and
- that it makes it possible to formally account for the success (or otherwise) of a robot by appealing to the notions of correct reasoning and correct representation.

This paper concerns the representation of knowledge about the objects in a robot's environment, and how such knowledge is acquired. The main feature of this knowledge is its incompleteness and uncertainty, placing the robot in what McCarthy calls the *common sense informatic situation* [1989]. The treatment given in the paper is rigorously logical, but has been carried through to implementation on a real robot.

1 ASSIMILATING SENSOR DATA

The key idea of this paper is to consider the process of assimilating a stream of sensor data as abduction. Given such a stream, the abductive task is to hypothesise the existence, shapes, and locations of objects which, given the output the robot has supplied to its motors, would explain that sensor data. This is, in essence, the map building task for a mobile robot.

More precisely, if a stream of sensor data is represented as the conjunction Ψ of a set of observation sentences, the task is to find an explanation of Ψ in the form of a logical description (a map) Δ_M of the initial locations and shapes of a number of objects, such that,

$$\Sigma_B \wedge \Sigma_E \wedge \Delta_N \wedge \Delta_M \vDash \Psi$$

where,

- Σ_B is a background theory, comprising axioms for change (including continuous change), action, space, and shape,
- Σ_E is a theory relating the shapes and movements of objects (including the robot itself) to the robot's sensor data, and
- Δ_N is a logical description of the movements of objects, including the robot itself.

The exact form of these components is described in the next three sections, which present formalisms for representing and reasoning about action, change, space, and shape. In practice, as we'll see, these components will have to be split into parts for technical reasons.

The provision of a logic-based theoretical account brings issues like noise and incompleteness into sharp focus, and permits their study within the same framework used to address wider epistemological questions in knowledge representation. It also enables the formal evaluation of algorithms for low-level motor-perception tasks by supplying a formalism in which these tasks can be precisely specified.

2 REPRESENTING ACTION

The formalism used in this paper to represent action and change, including continuous change, is adapted from the circumscriptive event calculus presented in [Shanahan, 1995b] However, it employs a novel solution to the frame problem, inspired by the work of Kartha and Lifschitz [1995]. The result is a considerable simplification of the formalism in [Shanahan, 1995b].

Throughout the paper, the language of many-sorted first-order predicate calculus with equality will be used, augmented with circumscription. Variables in formulae begin with lower-case letters and are universally quantified with maximum scope unless indicated otherwise.

In the event calculus, we have sorts for *fluents*, *actions* (or events), and *time points*. It's assumed that time points are interpreted by the reals, and that the usual comparative predicates, arithmetic functions, and trigonometric functions are suitably defined. The formula HoldsAt(f,t) says that fluent f is true at time point t. The formulae Initiates(a,f,t) and Terminates(a,f,t) say respectively that action a makes fluent f true from time point t, and that a makes f false from t. The effects of actions are described by a collection of formulae involving Initiates and Terminates.

For example, if the term Rotate(r) denotes a robot's action of rotating r degrees about some axis passing through its body, and the term Facing(r) is a fluent representing that the robot is facing in a direction r degrees from North, then we might write the following Initiates and Terminates formulae.

$$\text{Initiates(Rotate(r1),Facing(r2),t)} \leftarrow \text{HoldsAt(Facing(r3),t)} \wedge \text{r2} = \text{r3} + \text{r1} \quad (2.1)$$

$$\text{Terminates(Rotate(r1),Facing(r2),t)} \leftarrow \text{HoldsAt(Facing(r2),t)} \wedge \text{r1} \neq 0 \quad (2.2)$$

Once a fluent has been initiated or terminated by an action or event, it is subject to the common sense law of inertia, which is captured by the event calculus axioms to be presented shortly. This means that it retains its value (true or false) until another action or event occurs which affects that fluent.

A narrative of actions and events is described via the predicates Happens and Initially. The formula Happens(a,t) says that an action

---

[1] Department of Computer Science, Queen Mary & Westfield College, Mile End Road, London E1 4NS, England.

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W.Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

or event of type a occurred at time point t. Events are instantaneous. The formula Initially(f) says that the fluent f is true from time point 0. A theory will also include a pair of uniqueness-of-names axioms, one for actions and one for fluents.

The relationship between HoldsAt, Happens, Initiates, and Terminates is constrained by the following axioms. Note that a fluent does not hold at the time of an action or event that initiates it, but does hold at the time of an action or event that terminates it.

$$\text{HoldsAt}(f,t) \leftarrow \text{Initially}(f) \wedge \neg \text{Clipped}(0,f,t) \quad \text{(EC1)}$$

$$\begin{aligned}&\text{HoldsAt}(f,t2) \leftarrow \quad \text{(EC2)}\\ &\quad \text{Happens}(a,t1) \wedge \text{Initiates}(a,f,t1) \wedge t1 < t2 \wedge\\ &\qquad \neg \text{Clipped}(t1,f,t2)\end{aligned}$$

$$\begin{aligned}&\neg \text{HoldsAt}(f,t2) \leftarrow \quad \text{(EC3)}\\ &\quad \text{Happens}(a,t1) \wedge \text{Terminates}(a,f,t1) \wedge t1 < t2 \wedge\\ &\qquad \neg \text{Declipped}(t1,f,t2)\end{aligned}$$

$$\begin{aligned}&\text{Clipped}(t1,f,t2) \leftrightarrow \quad \text{(EC4)}\\ &\quad \exists\, a,t\ [\text{Happens}(a,t) \wedge\\ &\qquad [\text{Terminates}(a,f,t) \vee \text{Releases}(a,f,t)] \wedge t1 < t \wedge t < t2]\end{aligned}$$

$$\begin{aligned}&\text{Declipped}(t1,f,t2) \leftrightarrow \quad \text{(EC5)}\\ &\quad \exists\, a,t\ [\text{Happens}(a,t) \wedge\\ &\qquad [\text{Initiates}(a,f,t) \vee \text{Releases}(a,f,t)] \wedge t1 < t \wedge t < t2]\end{aligned}$$

These axioms introduce a new predicate Releases [Kartha & Lifschitz, 1994]. The formula Releases(a,f,t) says that action a exempts fluent f from the common sense law of inertia. This non-inertial status is revoked as soon as the fluent is initiated or terminated once more. The use of this predicate will be illustrated shortly in the context of continuous change.

Let the conjunction of (EC1) to (EC5) be denoted by EC. The circumscription policy to overcome the frame problem is the following. Given a conjunction of Happens and Initially formulae N, a conjunction of Initiates, Terminates and Releases formulae E, and a conjunction of uniqueness-of-names axioms U, we are interested in,

$$\begin{aligned}&\text{CIRC}[N ; \text{Happens}] \wedge\\ &\quad \text{CIRC}[E ; \text{Initiates, Terminates, Releases}] \wedge U \wedge EC\end{aligned}$$

This formula embodies a form of the common sense law of inertia, and thereby solves the frame problem. Further details of this solution are to be found in [Shanahan, 1996a]. The key to the solution is to put EC outside the scope of the circumscriptions, thus ensuring that the Hanks-McDermott problem is avoided [Hanks & McDermott, 1987]. In most cases, the two circumscriptions will yield predicate completions, making the overall formula manageable and intuitive.

3 DOMAIN CONSTRAINTS AND CONTINUOUS CHANGE

Two additional features of the calculus are important: the ability to represent domain constraints, and the ability to represent continuous change.

Domain constraints are straightforwardly dealt with in the proposed formalism. They are simply formulated as HoldsAt formulae with a single universally quantified time variable, and conjoined outside the scope of the circumscriptions along with EC. For example, the following domain constraint expresses the fact that the robot can only face in one direction at a time.

$$\text{HoldsAt}(\text{Facing}(r1),t) \wedge \text{HoldsAt}(\text{Facing}(r2),t) \rightarrow r1 = r2$$

In the event calculus, domain constraints are used to determine values for fluents that haven't been initiated or terminated by actions or events (non-inertial fluents) given the values of other fluents that have. (Domain constraints that attempt to constrain the relationship between inertial fluents can lead to inconsistency.)

Following [Shanahan, 1990], continuous change is represented through the introduction of a new predicate and the addition of an extra axiom. The formula Trajectory(f1,t,f2,d) represents that, if the fluent f1 is initiated at time t, then after a period of time d the fluent f2 holds. We have the following axiom.

$$\begin{aligned}&\text{HoldsAt}(f2,t2) \leftarrow \quad \text{(EC6)}\\ &\quad \text{Happens}(a,t1) \wedge \text{Initiates}(a,f1,t1) \wedge t1 < t2 \wedge\\ &\qquad t2 = t1 + d \wedge \text{Trajectory}(f1,t1,f2,d) \wedge\\ &\qquad\quad \neg \text{Clipped}(t1,f1,t2)\end{aligned}$$

Let CEC denote EC $\wedge$ (EC6), and U denote the conjunction of a set of uniqueness-of-names axioms. If R is the conjunction of a set of domain constraints and T is the conjunction of set of formulae constraining Trajectory, then we are interested in,

$$\begin{aligned}&\text{CIRC}[N ; \text{Happens}] \wedge\\ &\quad \text{CIRC}[E ; \text{Initiates, Terminates, Releases}] \wedge\\ &\qquad T \wedge R \wedge U \wedge CEC.\end{aligned}$$

Notice that we are at liberty to include formulae which describe triggered events in N. Here's an example of such a formula, which describes conditions under which the robot will collide with a wall lying on an East-West line 100 units north of the origin.

$$\begin{aligned}&\text{Happens}(\text{Bump},t) \leftarrow\\ &\quad \text{HoldsAt}(\text{Moving},t) \wedge \text{HoldsAt}(\text{Facing}(r),t) \wedge\\ &\qquad -90 < r < 90 \wedge \text{HoldsAt}(\text{Location}(\text{Robot},\langle x,90\rangle),t)\end{aligned}$$

4 REPRESENTING SPACE AND SHAPE

The formalism used in this paper to represent space and shape is adapted from [Shanahan, 1995a]. Space is considered a real-valued co-ordinate system. For present purposes we can take space to be the plane $\mathbb{R} \times \mathbb{R}$, reflecting the fact that the robot we will consider will move only in two dimensions. A *region* is a subset of $\mathbb{R} \times \mathbb{R}$. A *point* is a member of $\mathbb{R} \times \mathbb{R}$. I will consider only interpretations in which points are interpreted as pairs of reals, in which regions are interpreted as sets of points, and in which the $\in$ predicate has its usual meaning.

Objects occupy open, path-connected regions. For example, the following formula describes an open circle of radius z units centred on the origin.

$$p \in \text{Disc}(z) \leftrightarrow \text{Distance}(p,\langle 0,0\rangle) < z \quad \text{(Sp1)}$$

Distance is a function yielding a positive real number, defined in the obvious way.

$$\text{Distance}(\langle x1,y1\rangle,\langle x2,y2\rangle) = \sqrt{(x1-x2)^2 + (y1-y2)^2} \quad \text{(Sp2)}$$

The function Bearing is also useful.

$$\begin{aligned}&\text{Bearing}(\langle x1,y1\rangle,\langle x2,y2\rangle) = r \leftarrow \quad \text{(Sp3)}\\ &\quad z = \text{Distance}(\langle x1,y1\rangle,\langle x2,y2\rangle) \wedge z \neq 0 \wedge\\ &\qquad \text{Sin}(r) = \frac{x2-x1}{z} \wedge \text{Cos}(r) = \frac{y2-y1}{z}\end{aligned}$$

Using Distance and Bearing we can define a straight line as follows. The term Line(p1,p2) denotes the straight line whose end points are p1 and p2. The Line function is useful in defining shapes with straight line boundaries.

$$\begin{aligned}&p \in \text{Line}(p1,p2) \leftrightarrow \quad \text{(Sp4)}\\ &\quad \text{Bearing}(p1,p) = \text{Bearing}(p1,p2) \wedge\\ &\qquad \text{Distance}(p1,p) \leq \text{Distance}(p1,p2)\end{aligned}$$

Spatial occupancy is represented by the fluent Occupies. The term Occupies(w,g) denotes that object w occupies region g. No object can occupy two regions at the same time. This implies, for example, that if an object occupies a region g, it doesn't occupy any subset of g nor any superset of g. We have the following domain constraints.

$$[\mathrm{HoldsAt}(\mathrm{Occupies}(w,g1),t) \wedge \mathrm{HoldsAt}(\mathrm{Occupies}(w,g2),t)] \rightarrow g1 = g2 \quad \text{(Sp5)}$$

$$\mathrm{HoldsAt}(\mathrm{Occupies}(w1,g1),t) \wedge \mathrm{HoldsAt}(\mathrm{Occupies}(w2,g2),t) \wedge w1 \neq w2 \rightarrow \neg \exists p\, [p \in g1 \wedge p \in g2] \quad \text{(Sp6)}$$

The first of these axioms captures the uniqueness of an object's region of occupancy, and the second insists that no two objects overlap.

The term Displace(g,⟨x,y⟩) denotes the result of displacing the region g by x units east and y units north. The Displace function is primarily used to describe motion: if an object moves, the region it occupies is displaced.

$$\langle x1,y1\rangle \in \mathrm{Displace}(g,\langle x2,y2\rangle) \leftrightarrow \langle x1-x2,y1-y2\rangle \in g \quad \text{(Sp7)}$$

The final component of the framework is a means of default reasoning about spatial occupancy [Shanahan, 1995a]. Shortly, a theory of continuous motion will be described. This theory insists that, in order for an object to follow a trajectory in space, that trajectory must be clear. Accordingly, as well as capturing which regions of space are occupied, our theory of space and shape must capture which regions are unoccupied.

A suitable strategy is to make space empty by default. It's sufficient to apply this default just to the situation at time 0 — the common sense law of inertia will effectively carry it over to later times. The following axiom is required, which can be thought of as a *common sense law of spatial occupancy*.

$$\mathrm{AbSpace}(w) \leftarrow \mathrm{Initially}(\mathrm{Occupies}(w,g)) \quad \text{(Sp8)}$$

The predicate AbSpace needs to be minimised, with Initially allowed to vary.

Where previously we were interested in CIRC[N ; Happens], it's now convenient to split this circumscription into two, and to distribute Initially formulae in two places. Given,

- the conjunction O of Axioms (Sp1) to (Sp8),
- a conjunction M of Initially formulae which mention only the fluent Occupies, and
- a conjunction N of Happens formulae and Initially formulae which don't mention the fluent Occupies, and
- conjunctions E, T, R, U, and CEC as described in the last section,

we are now interested in,

$$\mathrm{CIRC}[O \wedge M ; \mathrm{AbSpace} ; \mathrm{Initially}] \wedge \mathrm{CIRC}[N ; \mathrm{Happens}] \wedge \mathrm{CIRC}[E ; \mathrm{Initiates}, \mathrm{Terminates}, \mathrm{Releases}] \wedge T \wedge R \wedge U \wedge \mathrm{CEC}.$$

5 SENSORS AND MOTORS: THE THEORY Σ_E

We now have the logical apparatus required to construct a formal theory of the relationship between a robot's motor activity, the world, and the robot's sensor data. The present paper assumes perfect motors and perfect sensors. The issue of noise is dealt with in [Shanahan, 1996b].

The robot used as an example throughout the rest of the paper is one of the simplest and cheapest commercially available mobile robotic platforms at the time of writing, namely the Rug Warrior described by Jones and Flynn [1993] (Figure 1). This is a small, wheeled robot with a 68000 series microprocessor plus 32K RAM on board. It has a very simple collection of sensors. These include three bump switches arranged around its circumference, which will be our main concern here. In particular, we will confine our attention to the two forward bump switches, which, in combination, can deliver three possible values for the direction of a collision.

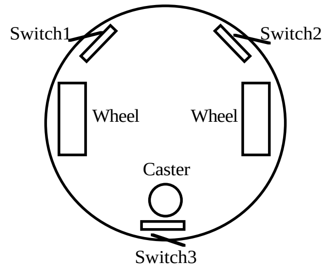

Figure 1: The Rug Warrior Robot from Above

Needless to say, each different kind of sensor gives rise to its own particular set of problems when it comes to constructing Σ_E. The question of noise is largely irrelevant when it comes to bump sensors. With infra-red proximity detectors, noise plays a small part. With sonar, the significance of noise is much greater. The use of cameras gives rise to a whole set of issues which are beyond the scope of this paper.

The central idea of this paper is the assimilation of sensor data through abduction. This is in accordance with the principle, "prediction is deduction but explanation is abduction" [Shanahan, 1989]. To begin with, we'll be looking at the predictive capabilities of the framework described.

The conjunction of our general theory of action, change, space, and shape with the theory Σ_E, along with a description of the initial locations and shapes of objects in the world and a description of the robot's actions, should yield a description of the robot's expected sensory input. If prediction works properly using deduction in this way, the reverse operation of explaining a given stream of sensor data by hypothesising the locations and shapes of objects in the world is already defined. It is simply abduction using the same logical framework.

In the caricature of the task of assimilating sensor data presented in Section 1, the relationship between motor activity and sensor data was described by Σ_E. In practice, this theory is split into parts and distributed across different circumscriptions (see Section 3).

First, we have a collection of formulae which are outside the scope of any circumscription. Let B be the conjunction of CEC with Axioms (B1) to (B6) below. Axioms (B1) and (B2) are uniqueness-of-names axioms. The robot is assumed to travel at a velocity of one unit of distance per unit of time.

$$\mathrm{UNA}[\mathrm{Occupies}, \mathrm{Facing}, \mathrm{Moving}, \mathrm{Blocked}, \mathrm{Touching}] \quad \text{(B1)}$$

$$\mathrm{UNA}[\mathrm{Rotate}, \mathrm{Go}, \mathrm{Stop}, \mathrm{Bump}, \mathrm{Switch1}, \mathrm{Switch2}] \quad \text{(B2)}$$

$$\mathrm{Trajectory}(\mathrm{Moving},t,\mathrm{Occupies}(\mathrm{Robot},g2),d) \leftarrow \mathrm{HoldsAt}(\mathrm{Occupies}(\mathrm{Robot},g1),t) \wedge \mathrm{HoldsAt}(\mathrm{Facing}(r),t) \wedge g2 = \mathrm{Displace}(g1,\langle d.\mathrm{Sin}(r),d.\mathrm{Cos}(r)\rangle) \quad \text{(B3)}$$

$$\mathrm{HoldsAt}(\mathrm{Facing}(r1),t) \wedge \mathrm{HoldsAt}(\mathrm{Facing}(r2),t) \rightarrow r1=r2 \quad \text{(B4)}$$

$$\mathrm{HoldsAt}(\mathrm{Blocked}(w1,w2,r),t) \leftrightarrow \exists g1,g2\, [\mathrm{HoldsAt}(\mathrm{Occupies}(w1,g1),t) \wedge \mathrm{HoldsAt}(\mathrm{Occupies}(w2,g2),t) \wedge w1 \neq w2 \wedge \exists z1\, [z1 > 0 \wedge \forall z2\, [z2 \leq z1 \rightarrow \exists p\, [p \in g2 \wedge p \in \mathrm{Displace}(g1,\langle z2.\mathrm{Sin}(r),z2.\mathrm{Cos}(r)\rangle)]]]] \quad \text{(B5)}$$

$$\begin{aligned}&\mathrm{HoldsAt}(\mathrm{Touching}(w1,w2,p),t) \leftrightarrow\\&\quad \mathrm{HoldsAt}(\mathrm{Occupies}(w1,g1),t) \wedge\\&\quad\quad \mathrm{HoldsAt}(\mathrm{Occupies}(w2,g2),t) \wedge w1 \neq w2 \wedge\\&\quad\quad\quad \exists\, p1, p2\, [p \in \mathrm{Line}(p1,p2) \wedge p \neq p1 \wedge p \neq p2 \wedge\\&\quad\quad\quad\quad \forall\, p3\, [[p3 \in \mathrm{Line}(p1,p) \wedge p3 \neq p] \rightarrow\\&\quad\quad\quad\quad\quad p3 \in g1] \wedge\\&\quad\quad\quad\quad\quad\quad \forall\, p3\, [[p3 \in \mathrm{Line}(p,p2) \wedge p3 \neq p] \rightarrow\\&\quad\quad\quad\quad\quad\quad\quad p3 \in g2]]\end{aligned} \tag{B6}$$

The fluent Blocked(w1,w2,r) holds if object w1 cannot move any distance at all in direction r without overlapping with another object. The fluent Touching(w1,w2,p) holds if w1 and w2 are touching at point p. This is true if a straight line exists from p1 to p2 at a bearing r which includes a point p3 such that every point between p1 and p3 apart from p3 itself is in g1 and every point from p2 to p3 apart from p3 itself is in g2.

Next we have a collection of Initiates, Terminates, and Releases formulae. Let E be the conjunction of the following axioms (E1) to (E6). A Bump event occurs when the robot collides with something.

$$\begin{aligned}&\mathrm{Initiates}(\mathrm{Rotate}(r1),\mathrm{Facing}(r1+r2),t) \leftarrow\\&\quad \mathrm{HoldsAt}(\mathrm{Facing}(r2),t)\end{aligned} \tag{E1}$$

$$\begin{aligned}&\mathrm{Releases}(\mathrm{Rotate}(r1),\mathrm{Facing}(r2),t) \leftarrow\\&\quad \mathrm{HoldsAt}(\mathrm{Facing}(r2),t) \wedge r1 \neq 0\end{aligned} \tag{E2}$$

$$\mathrm{Initiates}(\mathrm{Go},\mathrm{Moving},t) \tag{E3}$$

$$\mathrm{Releases}(\mathrm{Go},\mathrm{Occupies}(\mathrm{Robot},g),t) \tag{E4}$$

$$\begin{aligned}&\mathrm{Terminates}(a,\mathrm{Moving},t) \leftarrow\\&\quad a = \mathrm{Stop} \vee a = \mathrm{Bump} \vee a = \mathrm{Rotate}(r)\end{aligned} \tag{E5}$$

$$\begin{aligned}&\mathrm{Initiates}(a,\mathrm{Occupies}(\mathrm{Robot},g),t) \leftarrow\\&\quad [a = \mathrm{Stop} \vee a = \mathrm{Bump}] \wedge \mathrm{HoldsAt}(\mathrm{Occupies}(\mathrm{Robot},g),t)\end{aligned} \tag{E6}$$

Now we have a collection of formulae concerning the narrative of actions and events we're interested in. This collection has two parts. Let N be N1 $\wedge$ N2. The first component part concerns triggered events. The events Switch1 and Switch2 occur when the robot's forward bump switches are tripped (see Figure 1). Let N1 be the conjunction of Axioms (H1) to (H3) below.

$$\begin{aligned}&\mathrm{Happens}(\mathrm{Bump},t) \leftarrow\\&\quad [\mathrm{HoldsAt}(\mathrm{Moving},t) \vee \mathrm{Happens}(\mathrm{Go},t)] \wedge\\&\quad\quad \mathrm{HoldsAt}(\mathrm{Facing}(r),t) \wedge\\&\quad\quad\quad \mathrm{HoldsAt}(\mathrm{Blocked}(\mathrm{Robot},w,r),t)\end{aligned} \tag{H1}$$

$$\begin{aligned}&\mathrm{Happens}(\mathrm{Switch1},t) \leftarrow\\&\quad \mathrm{Happens}(\mathrm{Bump},t) \wedge \mathrm{HoldsAt}(\mathrm{Facing}(r),t) \wedge\\&\quad\quad \mathrm{HoldsAt}(\mathrm{Occupies}(\mathrm{Robot},\mathrm{Displace}(\mathrm{Disc}(z),p1)),t) \wedge\\&\quad\quad\quad \mathrm{HoldsAt}(\mathrm{Touching}(\mathrm{Robot},w,p2),t) \wedge\\&\quad\quad\quad\quad r-90 \leq \mathrm{Bearing}(p1,p2) < r+12\end{aligned} \tag{H2}$$

$$\begin{aligned}&\mathrm{Happens}(\mathrm{Switch2},t) \leftarrow\\&\quad \mathrm{Happens}(\mathrm{Bump},t) \wedge \mathrm{HoldsAt}(\mathrm{Facing}(r),t) \wedge\\&\quad\quad \mathrm{HoldsAt}(\mathrm{Occupies}(\mathrm{Robot},\mathrm{Displace}(\mathrm{Disc}(z),p1)),t) \wedge\\&\quad\quad\quad \mathrm{HoldsAt}(\mathrm{Touching}(\mathrm{Robot},w,p2),t) \wedge\\&\quad\quad\quad\quad r-12 \leq \mathrm{Bearing}(p1,p2) < r+90\end{aligned} \tag{H3}$$

The term Occupies(Robot,Displace(Disc(z),p1)) is employed in Axioms (H2) and (H3) to obtain the centre p1 of the region occupied by the robot, which can be thought of as its *location*. Note that Axiom (H1) caters for occasions on which the robot attempts to move when it is already blocked, as well as for occasions on which the robot's motion causes it to collide with something. In the former case, an immediate Bump event occurs, and the robot accordingly moves no distance at all.

For present purposes, the Bump event is somewhat redundant. In Axioms (E5) and (E6) it could be replaced by Switch1 and Switch2 events, and in Axioms (H2) and (H3) it could be simplified away. But abolishing the Bump event would violate a basic principle of the present approach, according to which the assumption of an external world governed by certain physical laws, a world to which its sensors have imperfect access, is built in to the robot. The robot's task is to do its best to explain its sensor data in terms of a model of the physics governing that world. In any such model, incoming sensor data is the end of the line, causally speaking. In the physical world, it's not a sensor event that stops the robot but a collision with a solid object.

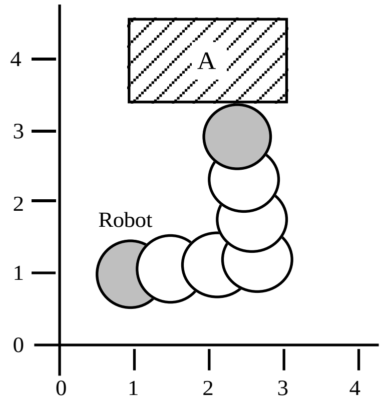

Figure 2: A Sequence of Robot Actions

The second component of N is a description of the robot's actions. Suppose the robot behaves as illustrated in Figure 2. Let N2 be the conjunction of the following formulae, which represent the robot's actions up to the moment when it bumps into obstacle A.

$$\mathrm{Happens}(\mathrm{Go},0) \tag{5.1}$$

$$\mathrm{Happens}(\mathrm{Stop},2{\cdot}8) \tag{5.2}$$

$$\mathrm{Happens}(\mathrm{Rotate}(-90),3{\cdot}3) \tag{5.3}$$

$$\mathrm{Happens}(\mathrm{Go},3{\cdot}8) \tag{5.4}$$

The final component of our theory is O $\wedge$ M, where M is a map of the robot's world and O is the conjunction of Axioms (Sp1) to (Sp8) Like N, M is conveniently divided into two parts. Let M be M1 $\wedge$ M2, where M1 is a description of the initial locations, shapes, and orientations (where applicable) of known objects, including the robot itself. For the example of Figure 2, M1 would be the conjunction of the following formulae.

$$\mathrm{Initially}(\mathrm{Facing}(80)) \tag{5.5}$$

$$\mathrm{Initially}(\mathrm{Occupies}(\mathrm{Robot},\mathrm{Displace}(\mathrm{Disc}(0{\cdot}5),\langle 1,1\rangle))) \tag{5.6}$$

The form of M2 is the same as that of M1. However, when assimilating sensor data, M2 is supplied by abduction. For now though, let's look at the predictive capabilities of this framework, and supply M2 directly. Let M2 be the following formula, which describes the obstacle in Figure 2.

$$\begin{aligned}&\exists\, g\, [\mathrm{Initially}(\mathrm{Occupies}(A,g)) \wedge\\&\quad \forall\, x,y\, [\langle x,y\rangle \in g \leftrightarrow 1 < x < 3 \wedge 3{\cdot}5 < y < 4{\cdot}5]]\end{aligned} \tag{5.7}$$

The following proposition says that, according to the formalisation, both bump switches are tripped at approximately time 5·5 (owing to a collision with obstacle A), and that the bump switches are not tripped at any other time.

Proposition 5.8.

$$\begin{aligned}&\mathrm{CIRC}\,[O \wedge M1 \wedge M2 \,;\, \mathrm{AbSpace} \,;\, \mathrm{Initially}] \wedge\\&\quad \mathrm{CIRC}[N1 \wedge N2 \,;\, \mathrm{Happens}] \wedge\\&\quad\quad \mathrm{CIRC}[E \,;\, \mathrm{Initiates}, \mathrm{Terminates}, \mathrm{Releases}] \wedge B \vDash\\&\quad\quad\quad \mathrm{Happens}(\mathrm{Switch1},T_{bump}) \wedge\\&\quad\quad\quad\quad \mathrm{Happens}(\mathrm{Switch2},T_{bump}) \wedge [[\mathrm{Happens}(\mathrm{Switch1},t) \vee\\&\quad\quad\quad\quad\quad \mathrm{Happens}(\mathrm{Switch2},t)] \rightarrow t = T_{bump}]\end{aligned}$$

where $T_{bump} = \frac{2\cdot 5 + 2\cdot 8.Cos(80)}{Cos(-10)} + 3\cdot 8$.

Proof. In full version of paper. □

The process of assimilating sensor data is the reverse of that of predicting sensor data. As outlined in Section 1, the task is to postulate the existence, location, and shape of a collection of objects which would explain the robot's sensor data, given its motor activity.

Let Ψ be the conjunction of a set of formulae of the form Happens(Switch1,τ) or Happens(Switch2,τ) where τ is a time point. What we want to explain is the *partial completion* of this formula, for reasons that will be made clear shortly. The only-if half of this completion is defined as follows.

Definition 5.9.

$$\mathrm{COMP}[\Psi] \equiv_{def}$$
$$[\mathrm{Happens}(a,t) \wedge [a = \mathrm{Switch1} \vee a = \mathrm{Switch2}]] \rightarrow \bigvee_{\langle\alpha,\tau\rangle \in \Gamma} [a = \alpha \wedge t = \tau]$$

where $\Gamma = \{\langle\alpha,\tau\rangle \mid \mathrm{Happens}(\alpha,\tau) \in \Psi\}$. □

Given Ψ, we're interested in finding conjunctions M2 of formulae in which each conjunct has the form,

$$\exists\, g\, [\mathrm{Initially}(\mathrm{Occupies}(\omega,g)) \wedge \forall\, p\, [p \in g \leftrightarrow \Pi]]$$

where ρ is a point constant, ω is an object constant, and Π is any formula in which p is free, such that O $\wedge$ M1 $\wedge$ M2 is consistent and,

$$\mathrm{CIRC}[O \wedge M1 \wedge M2 ; \mathrm{AbSpace} ; \mathrm{Initially}] \wedge$$
$$\mathrm{CIRC}[N1 \wedge N2 ; \mathrm{Happens}] \wedge$$
$$\mathrm{CIRC}[E ; \mathrm{Initiates, Terminates, Releases}] \wedge B \vDash$$
$$\Psi \wedge \mathrm{COMP}[\Psi].$$

The partially completed form of the Happens formula on the right-hand-side of the turnstile eliminates anomalous explanations in which, for example, the robot encounters a phantom extra obstacle before the time of the first event in Ψ. If Ψ on its own were used instead of this partially completed formula, it would be possible to construct such explanations by shifting all the obstacles that appear in a proper explanation into new positions which take account of the premature interruption in the robot's path caused by the phantom obstacle.

Clearly, from Proposition 5.8, if Ψ is,

$$\mathrm{Happens}(\mathrm{Switch1},T_{bump}) \wedge \mathrm{Happens}(\mathrm{Switch2},T_{bump})$$

then (5.7) is an explanation that meets this specification. Note that the symbol A in (5.7) (or rather its computational counterpart in the actual robot), when generated through the abductive assimilation of sensor data, is *grounded* in Harnad's sense of the term [Harnad, 1990], at the same time as acquiring *meaning* through the theory. Furthermore, the theoretical framework within which such explanations are understood,

- links the symbols that appear in them directly to a level of representation at which high-level reasoning tasks can be performed, and
- licenses an account of the robot's success (or otherwise) at performing its tasks which appeals to the correctness of its representations and its reasoning processes.

However, (5.7) is just one among infinitely many possible explanations of this Ψ of the required form. In the specification of an abductive task like this, the set of explanations of the required form will be referred to as the *hypothesis space*. It's clear, in the present case, that some constraints must be imposed on the hypothesis space to eliminate bizarre explanations. Furthermore, the set of all explanations of the suggested form for a given stream of sensor data is hard to reason about, and computing a useful representation of such a set is infeasible. This problem is tackled in the full paper by adopting a *boundary-based* representation of shape (see [Davis, 1990, Chapter 6]). Space limitations preclude further discussion of this topic here.

CONCLUDING REMARKS

A great deal of further work has already been completed, including a treatment of noise via non-determinism and a consistency-based form of abduction [Shanahan, 1996b]. This has led to the design of a provably correct algorithm for sensor data assimilation, which forms the basis of a C implementation which has been used in a number of experiments with the robot. All of this is described in the full paper, which is available from the author.

ACKNOWLEDGEMENTS

The inspiration for Cognitive Robotics comes from Ray Reiter and his colleagues at the University of Toronto. Thanks to Neelakantan Kartha and Rob Miller. The author is an EPSRC Advanced Research Fellow.

REFERENCES

[Davis, 1990] E.Davis, *Representations of Commonsense Knowledge*, Morgan Kaufmann (1990).

[Hanks & McDermott, 1987] S.Hanks and D.McDermott, Nonmonotonic Logic and Temporal Projection, *Artificial Intelligence*, vol 33 (1987), pages 379-412.

[Harnad, 1990] S.Harnad, The Symbol Grounding Problem, *Physica D*, vol 42 (1990), pages 335-346.

[Jones & Flynn, 1993] J.L.Jones and A.M.Flynn, *Mobile Robots: Inspiration to Implementation*, A.K.Peters (1993).

[Kartha & Lifschitz, 1994] G.N.Kartha and V.Lifschitz, Actions with Indirect Effects (Preliminary Report), *Proceedings 1994 Knowledge Representation Conference*, pages 341-350.

[Kartha & Lifschitz, 1995] G.N.Kartha and V.Lifschitz, A Simple Formalization of Actions Using Circumscription, *Proceedings IJCAI 95*, pages 1970-1975.

[Lespérance, *et al.*, 1994] Y.Lespérance, H.J.Levesque, F.Lin, D.Marcu, R.Reiter, and R.B.Scherl, A Logical Approach to High-Level Robot Programming: A Progress Report, in *Control of the Physical World by Intelligent Systems: Papers from the 1994 AAAI Fall Symposium*, ed. B.Kuipers, New Orleans (1994), pages 79-85.

[McCarthy, 1989] J.McCarthy, Artificial Intelligence, Logic and Formalizing Common Sense, in *Philosophical Logic and Artificial Intelligence*, ed. R.Thomason, Kluwer Academic (1989), pages 161-190.

[Shanahan, 1989] M.P.Shanahan, Prediction Is Deduction but Explanation Is Abduction, *Proceedings IJCAI 89*, pages 1055-1060.

[Shanahan, 1990] M.P.Shanahan, Representing Continuous Change in the Event Calculus, *Proceedings ECAI 90*, pages 598-603.

[Shanahan, 1995a] M.P.Shanahan, Default Reasoning about Spatial Occupancy, *Artificial Intelligence*, vol 74 (1995), pages 147-163.

[Shanahan, 1995b] M.P.Shanahan, A Circumscriptive Calculus of Events, *Artificial Intelligence*, vol 77 (1995), pages 249-284.

[Shanahan, 1996a] M.P.Shanahan, *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*, MIT Press (1996), to appear.

[Shanahan, 1996b] M.P.Shanahan, Noise and the Common Sense Informatic Situation for a Mobile Robot, *Proceedings AAAI 96*, to appear.