Robotics and the Common Sense Informatic Situation

Murray Shanahan

Department of Computer Science,
Queen Mary and Westfield College,
Mile End Road,
London E1 4NS,
England.
mps@dcs.qmw.ac.uk

Abstract

Any model of the world a robot constructs on the basis of its sensor data is necessarily both incomplete, due to the robot's limited window on the world, and uncertain, due to sensor and motor noise. This paper proposes a logic-based framework in which such models are constructed 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. Noise is treated as a kind of non-determinism, and is dealt with by a consistency-based form of abduction.

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 [Nilsson, 1984], 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 [Charniak & McDermott, 1985, page 455]. 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,

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

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.

Three major issues arise with this logical specification of the map building task: noisy data, incomplete information, and implementation.

- Σ_E does not have to assume a perfect correspondence between objects in the world and sensor data received from them, or a perfect correspondence between motor outputs and actual movements in the world. In practice, a noisy interface between world and robot must be assumed. Using the expressive power of first-order logic, the uncertainty resulting from such noise can be captured.

- Data in the common sense informatic situation is incomplete as well as noisy. In abductive terms, there will typically be many Δ_M's that could explain any given Ψ. For example, the robot may only receive sensor data from a small fraction of the total surface of an object, and be unable to tell whether the object is large or small. Again, using the expressive power of first-order logic, this incompleteness can be captured.
- This logical specification of the map building task must be rendered into an efficient implementation which can be executed by the on-board microprocessor of a mobile robot.

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], which in turn is based loosely on the formalism of Kowalski and Sergot [1986]. 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 [McCarthy, 1986], [Lifschitz, 1994]. 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.[1]

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

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

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 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. Here's an example narrative.

$$\text{Initially}(\text{Facing}(0)) \quad (2.3)$$

$$\text{Happens}(\text{Rotate}(90),10) \quad (2.4)$$

$$\text{Happens}(\text{Rotate}(-180),20) \quad (2.5)$$

A theory will also include a pair of uniqueness-of-names axioms, one for actions and one fluents.

$$\text{UNA}[\text{Facing}] \quad (2.6)$$

$$\text{UNA}[\text{Rotate}] \quad (2.7)$$

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.

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

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

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

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

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

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.

[1] Rotation is treated as instantaneous here, and throughout the sequel.

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,

CIRC[N ; Happens] ∧
  CIRC[E ; Initiates, Terminates, Releases] ∧ U ∧ EC

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, 1996]. 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.

For the example above, we have the following proposition. Let E be the conjunction of (2.1) with (2.2), let N be the conjunction of (2.3) to (2.5), and let U be the conjunction of (2.6) with (2.7).

Proposition 2.8.

CIRC[N ; Happens] ∧
  CIRC[E ; Initiates, Terminates, Releases] ∧ U ∧ EC ⊨
    HoldsAt(Facing(r),t) ←
      [0 ≤ t ≤ 10 ∧ r = 0] ∨ [10 < t ≤ 20 ∧ r = 90] ∨
        [20 < t ∧ r = 270].

Proof. See Appendix. □

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.

HoldsAt(Facing(r1),t) ∧ HoldsAt(Facing(r2),t) → 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.)[1]

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.

HoldsAt(f2,t2) ← (EC6)
  Happens(a,t1) ∧ Initiates(a,f1,t1) ∧ t1 < t2 ∧
    t2 = t1 + d ∧ Trajectory(f1,t1,f2,d) ∧
      ¬ Clipped(t1,f1,t2)

Let CEC denote EC ∧ (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,

CIRC[N ; Happens] ∧
  CIRC[E ; Initiates, Terminates, Releases] ∧
    T ∧ R ∧ U ∧ CEC.

For example, suppose the robot's repertoire of actions is expanded to include the actions Go and Stop. The Go action initiates a period of continuous change in the robot's location. The Stop action terminates such a period. The robot's location will be represented by the fluent Location(Robot,p), where p is a pair of Cartesian co-ordinates the form ⟨x,y⟩. (The first argument of this fluent is there so that we can represent the locations of other objects beside the robot. This will be useful later on.) A constant velocity V is assumed in the following collection of formulae, which are intended to capture this example.

Let E be the conjunction of the following formulae.

Initiates(Go,Moving,t) (3.1)

Releases(Go,Location(Robot,p),t) (3.2)

Terminates(Stop,Moving,t) (3.3)

Initiates(Stop,Location(Robot,p),t) ← (3.4)
  HoldsAt(Location(Robot,p),t)

Let T be the following formula.

Trajectory(Moving,t,Location(Robot,⟨x2,y2⟩),d) ← (3.5)
  HoldsAt(Location(Robot,⟨x1,y1⟩),t) ∧
    HoldsAt(Facing(r),t1) ∧
      x2 = x1 + V.d.Sin(r) ∧ y2 = y1 + V.d.Cos(r)

Let R be the following domain constraint.

[HoldsAt(Location(w,p1),t) ∧ (3.6)
  HoldsAt(Location(w,p2),t)] → p1 = p2

Let U be the conjunction of the following uniqueness-of-names axioms.

UNA[Location, Facing, Moving] (3.7)

UNA[Go, Stop] (3.8)

Let N be the following narrative description.

Initially(Location(Robot,⟨0,0⟩)) (3.9)

Initially(Facing(90)) (3.10)

Happens(Go,10) (3.11)

Happens(Stop,20) (3.12)

Now, given that the circumscriptions of E and N yield the predicate completions of Happens, Initiates, Terminates, and Releases, it's a straightforward exercise to

[1] Note that Initiates(a,F1,t) → Initiates(a,F2,t) does not follow from HoldsAt(F1,t) → HoldsAt(F2,t).

show that the recommended circumscription yields what we would expect.

Proposition 3.13.

CIRC[N ; Happens] $\wedge$
CIRC[E ; Initiates, Terminates, Releases] $\wedge$
$T \wedge R \wedge U \wedge CEC \models$
HoldsAt(Location(Robot,⟨x,y⟩),t) $\leftrightarrow$
$[0 \leq t \leq 10 \wedge x = 0 \wedge y = 0] \vee$
$[10 < t \leq 20 \wedge x = V.(t-10) \wedge y = 0] \vee$
$[20 < t \wedge x = V.10 \wedge y = 0]$.

Proof. See Appendix. □

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.

Happens(Bump,t) $\leftarrow$
HoldsAt(Moving,t) $\wedge$ HoldsAt(Facing(r),t) $\wedge$
$-90 < r < 90$ $\wedge$ HoldsAt(Location(Robot,⟨x,90⟩),t)

4 Representing Space and Shape

The formalism used in this paper to represent space and shape is taken 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.

A shape is represented as a region. The only shapes we will consider are open and path-connected. Every shape has a conventional centre, which is the origin ⟨0,0⟩.[1] For example, an open circle of radius z units is described by following formula.

$p \in$ Disc(z) $\leftrightarrow$ Distance(p,⟨0,0⟩) $< z$ (Sp1)

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

Distance(⟨x1,y1⟩,⟨x2,y2⟩) = $\sqrt{(x1-x2)^2 + (y1-y2)^2}$ (Sp2)

The function Bearing is also useful.

Bearing(⟨x1,y1⟩,⟨x2,y2⟩) = r $\leftarrow$ (Sp3)
z = Distance(⟨x1,y1⟩,⟨x2,y2⟩) $\wedge z \neq 0 \wedge$
$\mathrm{Sin}(r) = \frac{x2-x1}{z} \wedge \mathrm{Cos}(r) = \frac{y2-y1}{z}$

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.

[1] This conventional "centre" is just a reference point, and doesn't even have to be inside the shape in question.

$p \in$ Line(p1,p2) $\leftrightarrow$ (Sp4)
Bearing(p1,p) = Bearing(p1,p2) $\wedge$
Distance(p1,p) $\leq$ Distance(p1,p2)

Space is occupied by objects. Each object w has a unique shape denoted by the term Shape(w). If the robot is denoted by the term Robot, and if its body is circular and ten units in radius, then we can express this as follows.

Shape(Robot) = Disc(0·5)

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.

[HoldsAt(Occupies(w,g1),t) $\wedge$ (Sp5)
HoldsAt(Occupies(w,g2),t)] $\rightarrow$ g1 = g2

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

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

An object's location is represented by the fluent Location. A further domain constraint is required which relates Location to Occupies. The term Location(w,p), which we've already encountered, denotes that the object w is located at point p. This means that the region it occupies is the result of displacing the conventional centre of its shape by x units east and y units north, where p = ⟨x,y⟩. If the object's shape is the region g, then the result of this displacement is denoted by the term Displace(g,p).

HoldsAt(Occupies(w,Displace(g,p)),t) $\leftrightarrow$ (Sp7)
$\exists$ g [Shape(w,g) $\wedge$ HoldsAt(Location(w,p),t)]

⟨x1,y1⟩ $\in$ Displace(g,⟨x2,y2⟩) $\leftrightarrow$ ⟨x1−x2,y1−y2⟩ $\in$ g (Sp8)

Using the Displace function, shapes can be conveniently combined to form new shapes by taking their union (via a disjunction). The following formula defines a shape a little like the field of view through a pair of binoculars, formed from two overlapping circles.

$p \in$ TwoDiscs(x) $\leftrightarrow$
$p \in$ Displace(Disc(x),⟨$-\frac{x}{2}$,0⟩) $\vee$
$p \in$ Displace(Disc(x),⟨$\frac{x}{2}$,0⟩)

The incorporation of rotations in this formalism is extremely straightforward. In the present context, however, the only moving objects we'll encounter are circular, so the possibility of rotating a shape has been ignored.

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 for now 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*.

AbSpace(w) $\leftarrow$ Initially(Location(w,p)) (Oc1)

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) with Axiom (Oc1),
- a conjunction M of Initially formulae which mention only the spatial fluents Location and Occupies, and
- a conjunction N of Happens formulae and Initially formulae which don't mention the spatial fluents Location and Occupies, and
- conjunctions E, T, R, U, and CEC as described in the last section,

we are now interested in,

CIRC [O $\wedge$ M ; AbSpace ; Initially] $\wedge$
CIRC[N ; Happens] $\wedge$
CIRC[E ; Initiates, Terminates, Releases] $\wedge$
T $\wedge$ R $\wedge$ U $\wedge$ 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. For now we will assume perfect motors and perfect sensors. The issue of noise is dealt with in Section 7.

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 5a). 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.

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.

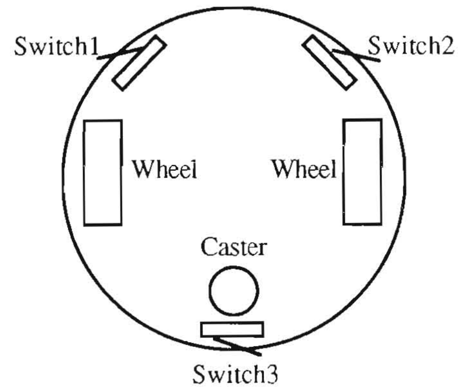

Figure 5a:
The Rug Warrior Robot from Above

In the caricature of the task of assimilating sensor data presented in Section 1, the realtionship 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 (B5) below. The robot is assumed to travel at a velocity of one unit of distance per unit of time.

UNA[Occupies, Location, Facing, Moving] (B1)

UNA[Rotate, Go, Stop] (B2)

Trajectory(Moving,t,Location(Robot,$\langle$x2,y2$\rangle$),d) $\leftarrow$ (B3)
HoldsAt(Location(Robot,$\langle$x1,y1$\rangle$),t) $\wedge$
HoldsAt(Facing(r),t) $\wedge$
x2 = x1 + d.Sin(r) $\wedge$ y2 = y1 + d.Cos(r)

HoldsAt(Blocked(w1,w2,r),t) $\leftrightarrow$ (B4)
HoldsAt(Occupies(w1,g1),t) $\wedge$
HoldsAt(Occupies(w2,g2),t) $\wedge$ w1 $\neq$ w2 $\wedge$
HoldsAt(Location(w1,p1),t) $\wedge$
$\neg\exists$ z1 [z1 > 0 $\wedge$ $\forall$ z2 [z2 $\leq$ z1 $\wedge$
Bearing(p1,p2,r) $\wedge$ Distance(p1,p2,z2) $\rightarrow$
$\neg\exists$ p [p $\in$ g2 $\wedge$ p $\in$ Displace(g1,p2)]]]

$$\begin{array}{l}\text{HoldsAt(Touching(w1,w2,p),t)} \leftrightarrow \\ \quad \text{HoldsAt(Occupies(w1,g1),t)} \wedge \\ \quad\quad \text{HoldsAt(Occupies(w2,g2),t)} \wedge \text{w1} \neq \text{w2} \wedge \\ \quad\quad\quad \exists \text{ p1, p2 [p} \in \text{Line(p1,p2)} \wedge \text{p} \neq \text{p1} \wedge \text{p} \neq \text{p2} \wedge \\ \quad\quad\quad\quad \forall \text{ p3 [[p3} \in \text{Line(p1,p)} \wedge \text{p3} \neq \text{p]} \rightarrow \\ \quad\quad\quad\quad\quad \text{p3} \in \text{g1]} \wedge \\ \quad\quad\quad\quad\quad\quad \forall \text{ p3 [[p3} \in \text{Line(p,p2)} \wedge \text{p3} \neq \text{p]} \rightarrow \\ \quad\quad\quad\quad\quad\quad\quad \text{p3} \in \text{g2]].}\end{array} \quad \text{(B5)}$$

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.

$$\begin{array}{l}\text{Initiates(Rotate(r1),Facing(r1+r2),t)} \leftarrow \\ \quad \text{HoldsAt(Facing(r2),t)}\end{array} \quad \text{(E1)}$$

$$\begin{array}{l}\text{Releases(Rotate(r1),Facing(r2),t)} \leftarrow \\ \quad \text{HoldsAt(Facing(r2),t)} \wedge \text{r1} \neq 0\end{array} \quad \text{(E2)}$$

$$\text{Initiates(Go,Moving,t)} \quad \text{(E3)}$$

$$\text{Releases(Go,Location(Robot,p),t)} \quad \text{(E4)}$$

$$\begin{array}{l}\text{Terminates(a,Moving,t)} \leftarrow \\ \quad \text{a = Stop} \vee \text{a = Bump} \vee \text{a = Rotate(r)}\end{array} \quad \text{(E5)}$$

$$\begin{array}{l}\text{Initiates(a,Location(Robot,p),t)} \leftarrow \\ \quad \text{[a = Stop} \vee \text{a = Bump]} \wedge \\ \quad\quad \text{HoldsAt(Location(Robot,p),t)}\end{array} \quad \text{(E6)}$$

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 5a). Let N1 be the conjunction of Axioms (H1) to (H3) below.[1]

$$\begin{array}{l}\text{Happens(Bump,t)} \leftarrow \\ \quad \text{[HoldsAt(Moving,t)} \vee \text{Happens(Go,t)]} \wedge \\ \quad\quad \text{HoldsAt(Facing(r),t)} \wedge \\ \quad\quad\quad \text{HoldsAt(Blocked(Robot,w,r),t)}\end{array} \quad \text{(H1)}$$

$$\begin{array}{l}\text{Happens(Switch1,t)} \leftarrow \\ \quad \text{Happens(Bump,t)} \wedge \text{HoldsAt(Facing(r),t)} \wedge \\ \quad\quad \text{HoldsAt(Location(Robot,p1),t)} \wedge \\ \quad\quad\quad \text{HoldsAt(Touching(Robot,w,p2),t)} \wedge \\ \quad\quad\quad\quad \text{r}-90 \leq \text{Bearing(p1,p2)} < \text{r}+12\end{array} \quad \text{(H2)}$$

$$\begin{array}{l}\text{Happens(Switch2,t)} \leftarrow \\ \quad \text{Happens(Bump,t)} \wedge \text{HoldsAt(Facing(r),t)} \wedge \\ \quad\quad \text{HoldsAt(Location(Robot,p1),t)} \wedge \\ \quad\quad\quad \text{HoldsAt(Touching(Robot,w,p2),t)} \wedge \\ \quad\quad\quad\quad \text{r}-12 \leq \text{Bearing(p1,p2)} < \text{r}+90\end{array} \quad \text{(H3)}$$

[1] Both forward bump switches are tripped if the collision point is within approximately 12° of the robot's bearing.

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. One reason not to abolish the Bump event is that, in principle, a collision could occur without the attendant sensor event — if one of the bump switches were broken, say. Similarly, a sensor event could occur without a collision as its cause — if a rain drop were to momentarily short a connection, for example.

Another reason is that 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.

The second component of N is a description of the robot's actions. Suppose the robot behaves as illustrated in Figure 5b. 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.

$$\text{Happens(Go,0)} \quad (5.1)$$

$$\text{Happens(Stop,2·8)} \quad (5.2)$$

$$\text{Happens(Rotate(}-90\text{),3·3)} \quad (5.3)$$

$$\text{Happens(Go,3·8)} \quad (5.4)$$

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) with Axiom (Oc1). 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 5b, M1 would be the conjunction of the following formulae.

$$\text{Initially(Facing(80))} \quad (5.5)$$

$$\text{Initially(Location(Robot,}\langle 1,1\rangle\text{))} \quad (5.6)$$

$$\text{Shape(Robot) = Disc(0·5)} \quad (5.7)$$

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 5b.

$$\begin{array}{l}\text{Initially(Location(A,}\langle 2,4\rangle\text{))} \wedge \\ \quad \forall \text{ x, y [}\langle \text{x,y}\rangle \in \text{Shape(A)} \leftrightarrow -1 < \text{x} < 1 \wedge \\ \quad\quad -0·5 < \text{y} < 0·5\text{]}\end{array} \quad (5.8)$$

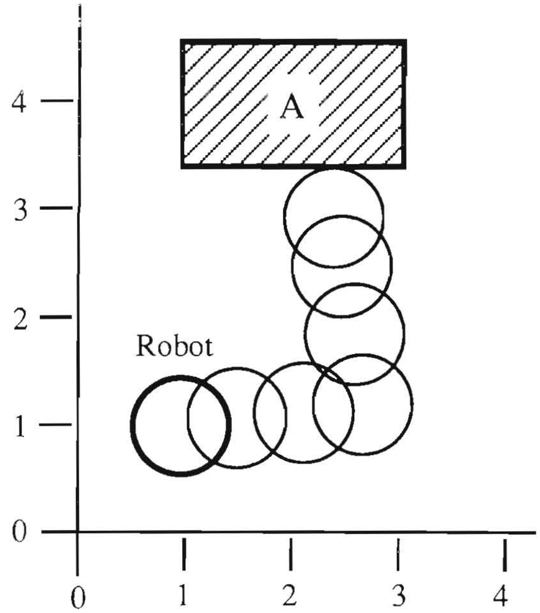

Figure 5b: A Sequence of Robot Actions

The following proposition says that, according to the formalisation, bump switch two is 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 5.9.

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

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

Proof. See Appendix. □

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.[1]

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 5.10.

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

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

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

$$\text{Initially}(\text{Location}(\omega,\rho)) \wedge \forall\, p\, [p \in \text{Shape}(\omega) \leftrightarrow \Pi]$$

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,

$$\begin{aligned}&\text{CIRC}[\text{O} \wedge \text{M1} \wedge \text{M2} \,;\, \text{AbSpace} \,;\, \text{Initially}] \wedge\\ &\quad \text{CIRC}[\text{N1} \wedge \text{N2} \,;\, \text{Happens}] \wedge\\ &\quad\quad \text{CIRC}[\text{E} \,;\, \text{Initiates, Terminates, Releases}] \wedge \text{B} \models\\ &\quad\quad\quad \Psi \wedge \text{COMP}[\Psi].\end{aligned}$$

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 interuption in the robot's path caused by the phantom obstacle.

Clearly, from Proposition 5.9, if Ψ is,

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

then (5.8) is an explanation that meets this specification.[2] Note that the symbol A in (5.8) (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, (5.8) is just one among infinitely many possible explanations of this Ψ of the required form. A bizarre example of an alternative explanation would be that the whole of space was occupied by a single object with a tunnel bored in it whose shape exactly matched that of the robot's path up to time T_{bump}.

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

[1] In the present paper, it is assumed that all sensor data require explanation. To take account of glitches (as opposed to just noise), this requirement can be relaxed.

[2] It is assumed that our language includes an arbitrarily large set of unused constant symbols, from which ω is drawn.

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.

6 Noise

The hallmark of the common sense informatic situation for a mobile robot is incomplete and uncertain knowledge of a spatially extended world of middle-sized objects. Incompleteness is a consequence of the robot's limited window on the world, and uncertainty results from noise in its sensors and actuators. This section deals with noise.

Both noisy sensors and noisy actuators can be captured using non-determinism. (An alternative is to use probability [Bacchus, *et al.*, 1995]). Here we'll only look at the uncertainty in the robot's location that results from its noisy motors. The robot's motors are "noisy" for various reasons. For example, the two wheels might rotate at slightly different speeds when the robot is trying to travel in a straight line, or the robot might be moving on a slope or a slippery surface. Motor noise of this kind can be captured using a non-deterministic Trajectory formula, such as the following replacement for Axiom (B3).[1]

$$
\begin{array}{l}
\exists\, x1, y1\ [\text{Trajectory}(\text{Moving},t, \qquad\qquad \text{(B6)}\\
\quad \text{Location}(\text{Robot},\langle x1,y1\rangle),d) \wedge\\
\quad\quad \text{Distance}(\langle x1,y1\rangle,\langle x2,y2\rangle) \leq d.\varepsilon] \leftarrow\\
\quad\quad\quad \text{HoldsAt}(\text{Location}(\text{Robot},\langle x3,y3\rangle),t) \wedge\\
\quad\quad\quad\quad \text{HoldsAt}(\text{Facing}(r),t) \wedge\\
\quad\quad\quad\quad\quad x2 = x3 + d.\text{Sin}(r) \wedge y2 = y3 + d.\text{Cos}(r)
\end{array}
$$

In effect, Axiom (B6) constrains the robot's location to be within an ever-expanding *circle of uncertainty* centred on the location it would be in if its motors weren't noisy.[2] The constant ε determines the rate at which this circle grows. Axiom (B7) below ensures that there are no discontinuities in the robot's trajectory. Without this axiom the robot would be able to leap over any obstacle which didn't completely cover the circle of uncertainty for its location. The term Abs(d) denotes the absolute value of d.

$$
\begin{array}{l}
\text{Trajectory}(f,t,\text{Location}(x,p1),d1) \rightarrow \qquad\qquad \text{(B7)}\\
\quad \forall\, z\ [z > 0 \rightarrow \exists\, d\ \forall\, d2, p2\ [[d2 > 0 \wedge\\
\quad\quad \text{Abs}(d2-d1) < d \wedge\\
\quad\quad\quad \text{Trajectory}(f,t,\text{Location}(x,p2),d2)] \rightarrow\\
\quad\quad\quad\quad \text{Distance}(p1,p2) < z]]
\end{array}
$$

Figure 6a shows the robot exploring the corner of an obstacle. Figure 6b shows the evolution of the corresponding circle of uncertainty, highlighting the points where the robot changes direction.

Figure 6b is somewhat misleading, however. Consider Figure 6c. On the top left, the evolution of the circle of uncertainty for the robot's location is shown. To the right, three potential locations are shown for the three changes of direction. Although these locations all fall within the relevant circles of uncertainty, the robot could never get to the third location from the second. This is because, as depicted at the bottom of the figure, in any given model the circle of uncertainty for the robot's location at the end of a period of continuous motion can only be defined relative to its actual location at the start of that period. This can be verified by inspecting Axioms (B6) and (B7).

The relative nature of the evolution of the circle of uncertainty means that the robot can acquire a detailed knowledge of some area A1 of its environment, then move to another area A2 which is some distance from A1, and acquire an equally detailed knowledge of A2. The accumulated uncertainty entails only that the robot is uncertain of where A1 is relative to A2. This natural feature of the formalisation conforms with what we would intuitively expect given the robot's informatic situation.

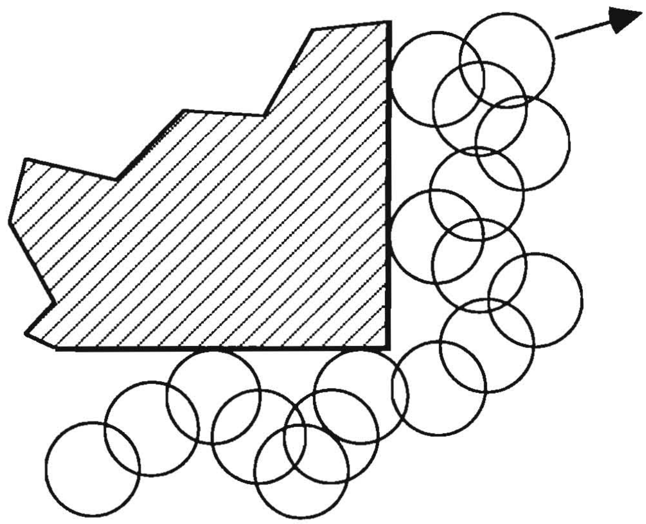

Figure 6a: The Robot Explores a Corner

Non-determinism is a potential source of difficulty for the abductive approach to explanation. Even with a precise and complete description of the initial state of the world, including all its objects and their shapes, a non-deterministic theory incorporating a formula like the one above will not yield the exact times at which collision events occur. Yet the sensor data that has to be assimilated has precise times attached to it. Fortunately we can recast the task of assimilating sensor data as a form of *weak abduction* so that it yields the required results. Intuitively what we want to capture is the fact that without the hypothesised objects, the sensor data could not have been received. This is analogous to the consistency-based approach to diagnosis proposed by Reiter [1987].

---

[1] The Rotate action could also be made non-deterministic.

[2] Note that, while objects occupy open subsets of $\mathbb{R}^2$, regions of uncertainty are closed.

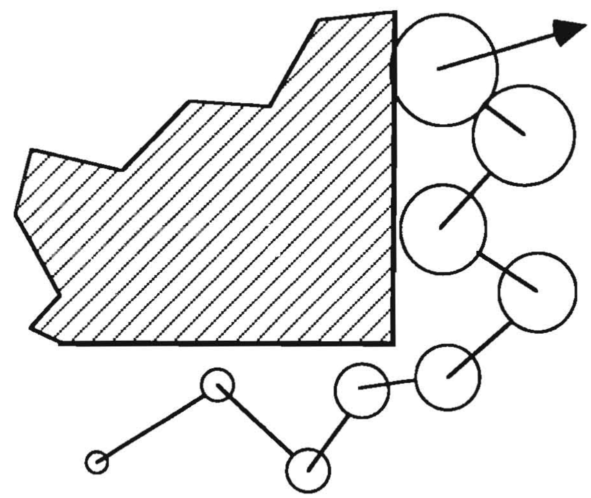

Figure 6b:
The Evolution of the Circle of Uncertainty

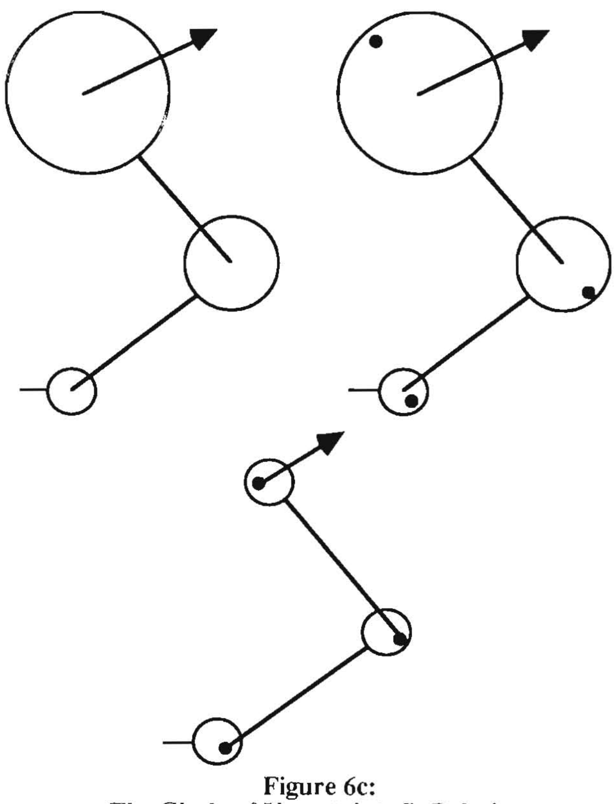

Figure 6c:
The Circle of Uncertainty Is Relative Not Absolute

Definition 6.1. Given,

- the conjunction B of CEC with Axioms (B1), (B2), and (B4) to (B7),
- the conjunction E of Axioms (E1) to (E6),
- the conjunction O of Axioms (Sp1) to (Sp8) with Axiom (Oc1),
- a conjunction M1 of Initially and Shape formulae describing the initial locations, shapes, and orientations of known objects, including the robot itself,
- the conjunction N1 of Axioms (H1) to (H3),
- a conjunction N2 of Happens formulae describing the robot's actions, and
- a conjunction Ψ of formulae of the form Happens(Switch1,τ) or Happens(Switch2,τ),

an *explanation* of Ψ is a conjunction M2 of formulae in which each conjunct has the form,

Initially(Location(ω,ρ)) $\wedge \forall$ p [p $\in$ Shape(ω) $\leftrightarrow \Pi$]

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,

CIRC[O $\wedge$ M1 $\wedge$ M2 ; AbSpace ; Initially] $\wedge$
CIRC[N1 $\wedge$ N2 ; Happens] $\wedge$
CIRC[E ; Initiates, Terminates, Releases] $\wedge$ B
$\nvDash \neg$ [$\Psi \wedge$ COMP[Ψ]]. □

There will, naturally, be many explanations for any given Ψ which meet this definition, even using the boundary-based representation of shape adopted in the full version of the paper. A standard way to treat multiple explanations in abductive knowledge assimilation is to adopt their disjunction. This has the effect of smothering any explanations which are stronger than necessary, such as those which postulate superfluous obstacles. The disjunction of all explanations of Ψ is the *cautious explanation* of Ψ.

A variety of *preference relations* over explanations can also be introduced. For example, it might be reasonable to assume that nearby collision points indicate the presence of a single object. Such preference relations are a topic for further study.

The following theorem establishes that the above definition of an explanation is equivalent to the deterministic specification offered in the last section when ε is 0. Let B_{det} be the conjunction of CEC with Axioms (B1) to (B5).

Definition 6.2. A formula M is a *complete spatial description* if the location and shape of every object mentioned in M is the same in every model of,

CIRC[O $\wedge$ M ; AbSpace ; Initially]. □

Theorem 6.3. If ε = 0 and M1 is a complete spatial description, then M2 is an explanation of Ψ if and only if O $\wedge$ M1 $\wedge$ M2 is consistent and,

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

Proof. See Appendix. □

A considerable amount of further work has been carried out, which is reported in the full version of the paper, but which it is only possible to present in outline here. Two further theorems have been established which characterise the abductive explanations defined above in terms which appeal more directly to the information available to any map-building algorithm which might be executed on board the robot. These theorems have been used to prove the correctness, with respect to the abductive specification given, of an algorithm for sensor data assimilation which constructs an *occupancy array* [Davis, 1990, Section 6.2.1].

This algorithm forms the core of an implementation in C, which runs on data acquired by the robot in the real world. Some preliminary experiments have been conducted in which the robot, under the control of a behaviour-based architecture [Brooks, 1986], explores an enclosure, and makes a record of its actions and sensor data for subsequent processing using the algorithm.

Concluding Remarks

In the paper accompanying his 1991 Computers and Thought Award Lecture, Brooks remarked that,

> [The field of Knowledge Representation] concentrates much of its energies on anomalies within formal systems which are never used for any practical task.
> [Brooks, 1991, page 578]

This paper should be construed as an answer to Brooks. According to the logical account given in this paper, a robot's incoming sensor data is filtered through an abductive process based on a framework of innate concepts, namely space, time, and causality. The development of a rigorous, formal account of this process bridges the gap between theoretical work in Knowledge Representation and practical work in robotics, and opens up a great many possibilities for further research. The following three issues are particularly pressing.

- The assimilation of sensor data from moving objects, such as humans, animals, or other robots. Movable obstacles should also be on the agenda.
- The assimilation of richer sensor data than that supplied by the Rug Warrior's simple bump switches.
- The control of the robot via the model of the world it acquires through abduction. Existing work in the Cognitive Robotics vein is likely to be influential here [Lespérance, *et al.*, 1994], [Kowalski, 1995], [Poole, 1995].

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

[Bacchus, *et al.*, 1995] F.Bacchus, J.Y.Halpren, and H.J.Levesque, Reasoning about Noisy Sensors in the Situation Calculus, *Proceedings IJCAI 95*, pages 1933-1940.

[Brooks, 1986] R.A.Brooks, A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, vol 2, no 1 (1986), pages 14-23.

[Brooks, 1991] R.A.Brooks, Intelligence Without Reason, *Proceedings IJCAI 91*, pages 569-595.

[Charniak & McDermott, 1985] E.Charniak and D.McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley (1985).

[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.

[Kowalski, 1995] R.A.Kowalski, Using Meta-Logic to Reconcile Reactive with Rational Agents, in *Meta-Logics and Logic Programming*, ed. K.R.Apt and F.Turini, MIT Press (1995), pages 227-242.

[Kowalski & Sergot, 1986] R.A.Kowalski and M.J.Sergot, A Logic-Based Calculus of Events, *New Generation Computing*, vol 4 (1986), pages 67-95.

[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 Syposium*, ed. B.Kuipers, New Orleans (1994), pages 79-85.

[Lifschitz, 1994] V.Lifschitz, Circumscription, in *The Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*, ed. D.M.Gabbay, C.J.Hogger and J.A.Robinson, Oxford University Press (1994), pages 297-352.

[McCarthy, 1986] J.McCarthy, Applications of Circumscription to Formalizing Common Sense Knowledge, *Artificial Intelligence*, vol 26 (1986), pages 89-116.

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

[Nilsson, 1984] N.J.Nilsson, ed., *Shakey the Robot*, SRI Technical Note no. 323 (1984), SRI, Menlo Park, California.

[Poole, 1995] D.Poole, Logic Programming for Robot Control, *Proceedings IJCAI 95*, pages 150-157.

[Reiter, 1987] R.Reiter, A Theory of Diagnosis from First Principles, *Artificial Intelligence*, vol 32 (1987), pages 57-95.

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

Appendix A: Proofs

Proof of Proposition 2.8. From CIRC[N ; Happens], we get,

$\text{Happens}(a,t) \leftrightarrow$ [A.1]
$[[a = \text{Rotate}(90) \wedge t = 10] \vee$
$[a = \text{Rotate}(-180) \wedge t = 20]]$.

From CIRC[E ; Initiates, Terminates, Releases], we get,

$\text{Initiates}(a,f,t) \leftrightarrow$ [A.2]
$a = \text{Rotate}(r1) \wedge f = \text{Facing}(r2) \wedge$
$\text{HoldsAt}(\text{Facing}(r3),t) \wedge r2 = r3 + r1$

$\text{Terminates}(a,f,t) \leftrightarrow$ [A.3]
$a = \text{Rotate}(r1) \wedge f = \text{Facing}(r2) \wedge$
$\text{HoldsAt}(\text{Facing}(r2),t) \wedge r1 \neq 0$

$\neg \exists a,f,t\ [\text{Releases}(a,f,t)]$. [A.4]

From [A.1] and (EC4), we get,

$\neg \text{Clipped}(0,\text{Facing}(0),t) \leftarrow 0 \leq t \leq 10$

which, given (EC1) and (2.3), yields,

$\text{HoldsAt}(\text{Facing}(0),t) \leftarrow 0 \leq t \leq 10$. [A.5]

From [A.2] and [A.5], we get,

$\text{Initiates}(\text{Rotate}(90),\text{Facing}(90),10)$. [A.6]

From [A.1] and (EC4), we get,

$\neg \text{Clipped}(10,\text{Facing}(90),t) \leftarrow 10 < t \leq 20$

which, given [A.1], [A.6] and (EC2), yields,

$\text{HoldsAt}(\text{Facing}(90),t) \leftarrow 0 \leq t \leq 10$. [A.7]

From [A.2] and [A.5], we get,

$\text{Initiates}(\text{Rotate}(-180),\text{Facing}(270),20)$. [A.8]

From [A.1] and (EC4), we get,

$\neg \text{Clipped}(20,\text{Facing}(270),t) \leftarrow 20 < t$

which, given [A.1], [A.8] and (EC2), yields,

$\text{HoldsAt}(\text{Facing}(270),t) \leftarrow 20 < t$. [A.9]

The proposition follows from [A.5], [A.7], and [A.9]. □

Proof of Proposition 3.13. From CIRC[N ; Happens], we get,

$\text{Happens}(a,t) \leftrightarrow$ [A.10]
$[[a = \text{Go} \wedge t = 10] \vee [a = \text{Stop} \wedge t = 20]]$.

From CIRC[E ; Initiates, Terminates, Releases], we get,

$\text{Initiates}(a,f,t) \leftrightarrow$ [A.11]
$[a = \text{Go} \wedge f = \text{Moving}] \vee$
$[a = \text{Stop} \wedge f = \text{Location}(\text{Robot},p) \wedge$
$\text{HoldsAt}(\text{Location}(\text{Robot},p),t)]$

$\text{Terminates}(a,f,t) \leftrightarrow a = \text{Stop} \wedge f = \text{Moving}$ [A.12]

$\text{Releases}(a,f,t) \leftrightarrow$ [A.13]
$a = \text{Go} \wedge f = \text{Location}(\text{Robot},p)$.

From [A.10] and (EC4), we get,

$\neg \text{Clipped}(0,\text{Location}(\text{Robot},\langle 0,0\rangle),t) \leftarrow 0 \leq t \leq 10$

which, from (EC1) and (3.9), yields,

$\text{HoldsAt}(\text{Location}(\text{Robot},\langle 0,0\rangle),t) \leftarrow 0 \leq t \leq 10$. [A.14]

Similarly, we can show,

$\text{HoldsAt}(\text{Facing}(90),10)$. [A.15]

From [A.11] we have,

$\text{Initiates}(\text{Go},\text{Moving},10)$. [A.16]

From [A.10] and (EC4), we get,

$\neg \text{Clipped}(10,\text{Moving},t) \leftarrow 10 < t \leq 20$. [A.17]

From [A.14], [A.15], and (3.5), we get,

$\text{Trajectory}(\text{Moving},10,\text{Location}(\text{Robot},\langle x,0\rangle),d) \leftarrow$
$x = V.d$

which, given [A.10], [A.16], [A.17] and (EC6), yields,

$\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,0\rangle),t) \leftarrow$ [A.18]
$10 < t \leq 20 \wedge x = V.(t-10)$.

From [A.11] and [A.18], we have,

$\text{Initiates}(\text{Stop},\text{Location}(\text{Robot},\langle x,0\rangle),20) \leftarrow$ [A.19]
$x = V.10$.

From [A.10] and (EC4), we get,

$\neg \text{Clipped}(20,\text{Location}(\text{Robot},\langle x,0\rangle),t) \leftarrow$
$20 < t \wedge x = V.10$

which, given [A.10], [A.18], [A.19] and (EC2), yields,

$\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,0\rangle),t) \leftarrow$ [A.20]
$20 < t \wedge x = V.10$.

From [A.14], [A.18] and [A.20], we arrive at,

$\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,y\rangle),t) \leftarrow$
$[0 \leq t \leq 10 \wedge x = 0 \wedge y = 0] \vee$
$[10 < t \leq 20 \wedge x = V.(t-10) \wedge y = 0] \vee$
$[20 < t \wedge x = V.10 \wedge y = 0]$.

The proposition follows from this and the domain constraint (3.6). □

Proof of Proposition 5.9. From CIRC[N1 $\wedge$ N2 ; Happens], we get,

$$\text{Happens}(a,t) \leftrightarrow \quad [A.21]$$
$$H1(a,t) \vee H2(a,t) \vee H3(a,t) \vee H4(a,t)$$

where,

$$H1(a,t) \equiv_{def}$$
$$[a = \text{Go} \wedge t = 0] \vee [a = \text{Stop} \wedge t = 2{\cdot}8] \vee$$
$$[a = \text{Rotate}(-90) \wedge t = 3{\cdot}3] \vee [a = \text{Go} \wedge t = 3{\cdot}8]$$

$$H2(a,t) \equiv_{def}$$
$$\exists\, w,r\ [a = \text{Bump} \wedge [\text{HoldsAt}(\text{Moving},t) \vee t = 3{\cdot}8] \wedge$$
$$\text{HoldsAt}(\text{Facing}(r),t) \wedge$$
$$\text{HoldsAt}(\text{Blocked}(\text{Robot},w,r),t)]$$

$$H3(a,t) \equiv_{def}$$
$$\exists\, w,r\ [a = \text{Switch1} \wedge [\text{HoldsAt}(\text{Moving},t) \vee t = 3{\cdot}8] \wedge$$
$$\text{HoldsAt}(\text{Facing}(r),t) \wedge$$
$$\text{HoldsAt}(\text{Location}(\text{Robot},p1),t) \wedge$$
$$\text{HoldsAt}(\text{Touching}(\text{Robot},w,p2),t) \wedge$$
$$r-150 < \text{Bearing}(p1,p2) < r+30]$$

$$H4(a,t) \equiv_{def}$$
$$\exists\, w,r\ [a = \text{Switch2} \wedge [\text{HoldsAt}(\text{Moving},t) \vee t = 3{\cdot}8] \wedge$$
$$\text{HoldsAt}(\text{Facing}(r),t) \wedge$$
$$\text{HoldsAt}(\text{Location}(\text{Robot},p1),t) \wedge$$
$$\text{HoldsAt}(\text{Touching}(\text{Robot},w,p2),t) \wedge$$
$$r-30 < \text{Bearing}(p1,p2) < r+150].$$

From CIRC[E ; Initiates, Terminates, Releases], we get,

$$\text{Initiates}(a,f,t) \leftrightarrow \quad [A.22]$$
$$[a = \text{Rotate}(r1) \wedge f = \text{Facing}(r1+r2) \wedge$$
$$\text{HoldsAt}(\text{Facing}(r2),t)] \vee [a = \text{Go} \wedge f = \text{Moving}] \vee$$
$$[[a = \text{Stop} \vee a = \text{Bump}(r)] \wedge$$
$$f = \text{Location}(\text{Robot},p) \wedge$$
$$\text{HoldsAt}(\text{Location}(\text{Robot},p),t)]$$

$$\text{Terminates}(a,f,t) \leftrightarrow \quad [A.23]$$
$$[a = \text{Stop} \vee a = \text{Bump}(r) \vee a = \text{Rotate}(r)] \wedge$$
$$f = \text{Moving}$$

$$\text{Releases}(a,f,t) \leftrightarrow \quad [A.24]$$
$$a = \text{Go} \wedge f = \text{Location}(\text{Robot},p).$$

From CIRC [O $\wedge$ M1 $\wedge$ M2 ; AbSpace ; Initially] we get,

$$\text{Initially}(\text{Location}(x,p)) \rightarrow x = A \vee x = \text{Robot}. \quad [A.25]$$

It can easily be shown that A retains its initial location for all time. Let $X_{turn} = 1 + 2{\cdot}8.\text{Sin}(80)$, $Y_{turn} = 1 + 2{\cdot}8.\text{Cos}(80)$. From [A.25] and (5.8), using Axioms (Sp7), (Sp8) and (B4), it can be confirmed that,

$$\neg\, \text{HoldsAt}(\text{Blocked}(\text{Robot},w,r),t) \leftarrow \quad [A.26]$$
$$\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,y\rangle),t) \wedge$$
$$\exists\, d\ [[0 \le d \le 2{\cdot}8 \wedge x = 1 + d.\text{Sin}(80) \wedge$$
$$y = 1 + d.\text{Cos}(80)] \vee$$
$$[3{\cdot}8 < d < T_{bump} \wedge$$
$$x = X_{turn} + (d - 3{\cdot}8).\text{Sin}(-10) \wedge$$
$$y = Y_{turn} + (d - 3{\cdot}8).\text{Cos}(-10)]].$$

It can similarly be confirmed that,

$$\neg\, \text{HoldsAt}(\text{Touching}(\text{Robot},w,p),t) \leftarrow \quad [A.27]$$
$$\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,y\rangle),t) \wedge$$
$$\exists\, d\ [[0 \le d \le 2{\cdot}8 \wedge x = 1 + d.\text{Sin}(80) \wedge$$
$$y = 1 + d.\text{Cos}(80)] \vee$$
$$[3{\cdot}8 < d < T_{bump} \wedge$$
$$x = X_{turn} + (d - 3{\cdot}8).\text{Sin}(-10) \wedge$$
$$y = Y_{turn} + (d - 3{\cdot}8).\text{Cos}(-10)]].$$

Given [A.26], from [A.21] to [A.24], using a similar procedure to that employed in the proof of Proposition 3.13, we can show,

$$\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,y\rangle),t) \leftarrow \quad [A.28]$$
$$[0 \le t \le 2{\cdot}8 \wedge x = 1 + t.\text{Sin}(80) \wedge$$
$$y = 1 + t.\text{Cos}(80)] \vee$$
$$[2{\cdot}8 < t \le 3.8 \wedge x = X_{turn} \wedge y = Y_{turn}] \vee$$
$$[3{\cdot}8 < t \le T_{bump} \wedge$$
$$x = X_{turn} + (t - 3{\cdot}8).\text{Sin}(-10) \wedge$$
$$y = Y_{turn} + (t - 3{\cdot}8).\text{Cos}(-10)].$$

Given that A retains its initial location, from [A.28], [A.25] and (5.8), using Axioms (Sp7) and (Sp8), we can show,

$$\text{HoldsAt}(\text{Blocked}(\text{Robot},A,-10),T_{bump}). \quad [A.29]$$

We can also show,

$$\text{HoldsAt}(\text{Facing}(-10),T_{bump}). \quad [A.30]$$

From [A.29] and [A.30], using Axiom (B4), we get,

$$\text{Happens}(\text{Bump},T_{bump}). \quad [A.31]$$

Given that A retains its initial location, from [A.28], [A.25] and (5.8), using Axioms (Sp7) and (Sp8), we can show,

$$\exists\, p1, p2\ [\quad [A.32]$$
$$\text{HoldsAt}(\text{Touching}(\text{Robot},A,p1),T_{bump}) \wedge$$
$$\text{HoldsAt}(\text{Location}(\text{Robot},p2),T_{bump}) \wedge$$
$$\text{Bearing}(p1,p2) = 0].$$

From [A.21] and [A.30] to [A.32] we get,

$$\text{Happens}(\text{Switch1},T_{bump}) \wedge \quad [A.33]$$
$$\text{Happens}(\text{Switch2},T_{bump}).$$

From [A.21], [A.27] and Axiom (B5) we get,

$$[\text{Happens}(\text{Switch1},t) \vee \text{Happens}(\text{Switch2},t)] \rightarrow \quad [A.34]$$
$$t = T_{bump}.$$

The proposition follows directly from [A.33] and [A.34]. □

Proof of Theorem 7.5. We only need to consider Ψ since the definition of an explanation caters for COMP[Ψ] automatically. The theorem follows from the fact that Axioms (B3) and (B5) are equivalent if ε is 0, and the fact that (B3) ensures that the robot's path is deterministic in the sense that at any given time its location is the same in every model of,

$$\text{CIRC}[O \wedge M1 \wedge M2 \text{ ; AbSpace ; Initially}] \wedge$$
$$\text{CIRC}[N1 \wedge N2 \text{ ; Happens}] \wedge$$
$$\text{CIRC}[E \text{ ; Initiates, Terminates, Releases}] \wedge B_{det}.$$

To see that the theorem follows, consider that, if the robot's path is deterministic according to a formula Γ and the locations and shapes of objects are the same in every model of Γ (as they must be in the above formula since M1 and M2 are complete spatial descriptions), then $\Gamma \not\models \neg\, \Psi$ if and only if $\Gamma \models \Psi$. □