Towards a formalization of the Spatial Semantic Hierarchy

Emilio Remolina and Benjamin Kuipers
Computer Science Department
University of Texas at Austin
Austin, Texas 78712 USA
e-mail: {eremolin,kuipers}@cs.utexas.edu

December 10, 1997

Abstract

The Spatial Semantic Hierarchy (SSH) comprises a set of distinct representations of space, each with its own ontology, each with its own mathematical foundation, and each abstracted from the levels below it. In particular, the SSH topological level is abstracted from the SSH causal level. The method used to abstract the SSH topological level has usually been defined as an abduction task, described in procedural terms according to the current implementation of the SSH. In this paper we define the circumscriptive theories associated with the SSH causal and topological levels. These theories are used to formalize the SSH abduction task and prove our implementation correct. Moreover, these theories show how topological information is used to dictate spatial distinctions that cannot be derived from causal information alone.

1 Introduction

The Spatial Semantic Hierarchy (SSH) is a set of distinct representations for space, each with its own ontology, each with its own mathematical foundation, and each abstracted from the levels below it. At the control level, the robot and its environment are modeled as a continuous dynamical system, whose stable equilibrium points are abstracted to a discrete set of "distinctive states". The control laws whose execution defines trajectories linking these states can be abstracted to actions, giving a discrete causal graph level of representation for the state space. Depending on the properties of the actions, the causal graph can be deterministic or stochastic. The causal graph of states and actions can in turn be abstracted to a topological network of places and paths (i.e. the topological map). Local metrical models, such as occupancy grids, of neighborhoods of places and paths can then be built on the framework of the topological network while avoiding their usual problems of global consistency.

The construction of the topological map is usually described ([1, 2, 3]) as an abduction process. This abduction task is defined in procedural terms inspired by the current implementation of the SSH. In this paper we define the circumscriptive theories associated with the SSH causal and topological levels. The SSH abduction task is then seen as constructing a model for these theories. Moreover, these theories show how topological information can be used to distinguish distinctive states that cannot be distinguished by using causal information alone.

The paper is organized as follows. Section 2 reviews the main aspects of the SSH. Section 3 defines the circumscriptive theories associated with the SSH causal and topological levels. We provide examples illustrating the interplay between the topological and causal graphs as well as some considerations when defining the circumscriptive policy associated with these theories. Section 4 presents some related work in the areas of automata learning [4, 5] and cognitive robotics [6, 7]. Finally, section 5 presents our conclusions and future work.

2 The Spatial Semantic Hierarchy

The Spatial Semantic Hierarchy (SSH) ([1, 2, 3, 8])[1] is an *ontological hierarchy* of representations for knowledge of large-scale space [2]. An ontological hierarchy shows how multiple representations for the same kind of knowledge can coexists. Each level of the hierarchy has its own *ontology* (the set of objects and relations it uses for describing the world) and its own set of inference and problem-solving methods. The objects, relations, and assumptions required by each level are provided by those below it.

[1]This presentation follows [3]

[2]In large-scale space the structure of the environment is revealed by integrating local observations over time, rather than being perceived from a single vantage point

The SSH abstracts the structure of an agent's spatial knowledge in a way that is relatively independent of its sensorimotor apparatus and the environment within which it moves. Next we describe the different SSH levels.

- The *sensorimotor level* of the agent provides continous sensors and effectors, but not direct access to the global structure of the environment, or the robot's position or orientation within it.

- At the *control level* of the hierarchy, the ontology is an egocentric sensorimotor one, without knowledge of fixed objects or places in an external environment. A *distinctive state* is defined as the local maximum found by a hill-climbing control strategy, climbing the gradient of a selected feature, or *distinctiveness measure*. Trajectory-following control laws take the robot from one distinctive state to the neighborhood of the next, where hill-climbing can find a local maximum, reducing position error and preventing its accumulation.

- The ontology at the SSH *causal level* consists of views, distinctive states, actions and schemas. A *view* is a description of the sensory input obtained at a locally distinctive state. An *action* denotes a sequence of one or more control laws which can be initiated at a locally distinctive state, and terminates after a hill climbing control law with the robot at another distinctive state. A *schema* is a tuple $((V, dp), A, (V', dq))$ representing the (temporally extended) event in which the robot takes a particular action A, starting with view V at the distinctive state dp, and terminating with view V' at distinctive state dq. In addition, we require that $dp \neq dq$[3].

- At the *topological level* of the hierarchy, the ontology consists of *places,paths and regions*, with connectivity and containment relations. Relations among the distinctive states and trajectories defined by the control level, and among their summaries as schemas at the causal level, are effectively described by the topological network. This network can be used to guide exploration of new environments and to solve new route-finding problems. Using the network representation, navigation among distinctive states is not dependent on the accuracy, or even the existence, of metrical knowledge of the environment.

- At the *metrical level* of the hierarchy, the ontology for places, paths, and sensory features is extended to include metrical properties such as distance, direction, shape, etc. Geometrical features are extracted from sensory input, and represented as annotations on the places and paths of the topological network.

Two fundamental ontological distinctions are embeded in the SSH. First, the continuous world of the sensorimotor and control levels is abstracted to the discrete symbolic representation at the causal and topological levels, to which the metrical level adds continuous properties. Second, the egocentric world of the sensorimotor, control, and causal level is abstracted to the world-centered ontologies of the topological and metrical levels.

[3]For example, we do not allow turns of 360 degrees.

Formalizing the levels of the hierarchy draws on different bodies of relevant theory: the sensorimotor and control levels on control theory and dynamical systems; the causal level on logic and stochastic transition models; the topological level on logic and simple topology; the geometrical level on estimation theory and differential geometry.

3 Formalizing the SSH Causal and Topological levels

As the agent navigates its environment, a set of schemas summarizing its experiences is created (see below). This set of schemas is the only source of information the agent has to create a spatial representation of its environment. At the causal level, the spatial representation posits the minimal set of distinctive states consistent with the set of schemas. At the topological level, the spatial representation posits the minimal set of paths and places consistent with the set of schemas. In order to meet these minimality conditions, the causal and topological levels are formalized as circumscriptive theories. Before defining these theories, we describe how a set of schemas is created as the agent navigates through its environment using continous control laws ([9]).

3.1 Creating Schemas

At the SSH control level, exploration is performed by alternating execution between two types of continous control strategies, *trajectory-following* and *hill-climbing* . These two types of control strategy differ in their roles: a *hill-climbing* control strategy is for climbing towards a local maximum of a distinctiveness measure and thus a position of some distinctive state; a *trajectory-following* control strategy is for moving from the neighborhood of one distinctive state to the neighborhood of another. The actual motion from one distinctive state to the neighborhood of another may be the result of the execution of a sequence of more than one local control strategy (see example below). Let $cl = cl_1, \ldots, cl_m$ be the sequence of control strategies executed at the control level to take the agent from distinctive state dp with view V to distinctive state dq with view V'. Then, the schema $((V, dp), A, (V', dq))$ is created at the SSH causal level, where A is an *action* symbol used whenever the sequence cl is executed. This way, the experiences of the robot within its environment can be described by an alternating sequence of views and actions

$$(V_1, d_1)\, A_1\, (V_2, d_2)\, \ldots A_{n-1}\, (V_n, d_n)$$

which is summarized at the causal level by the set of schemas S,

$$S = \{((V_i, d_i), A_i, (V_{i+1}, d_{i+1})) : i = 1, \ldots, n-1\}$$

At the SSH topological level, action symbols are categorized in two classes: *Turn* and *Travel*. *Turn* actions are associated with sequences of control strategies that move the agent from one distinctive state to another, where the initial and final distinctive state have the same position but different orientations. *Travel* actions are sequences of control strategies that move the agent to a different position in the environment.

The next example illustrates how sequences of control strategies at the control level are abstracted to schemas at the causal level.

Example 1 *Consider the environment in figure 1. In order to go from distinctive state d1 to distinctive state d2, the agent executes the sequence of control strategies* <`get-into-corridor, follow-middle-line, hc-T-intersection`> *where get-into-corridor is a trajectory-following control strategy that moves the agent from d1 to point a,* `follow-middle-line` *is a trajectory-following strategy that takes the agent from point a to point b, and* `hc-T-intersection` *is a hill-climbing control strategy that takes the agent from point b to the distinctive state d2.*[4] *At the distinctive state d2 the agent is facing the wall ahead and it is equidistant from this wall and the intersection corners.*

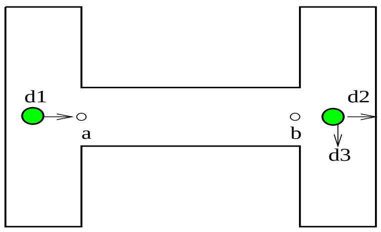

Figure 1: A sequence of control strategies , <*get-into-corridor, follow-middle-line, hc-T-intersection*>, takes the agent from distinctive state $d1$ to distinctive state $d2$. At the causal level, this continuous motion is represented by the schema $((V, d1), A, (V', d2))$, where the action symbol A represents the sequence <*get-into-corridor, follow-middle-line, hc-T-intersection*>, V and V' are the views at $d1$ and $d2$ respectively.

Distinctive state d3 is at the same physical location as d2 but with a different orientation. When the robot is at d3, it is facing the open space (corridor) at the right of d2. In order to go from distinctive state d2 to distinctive state d3, the agent executes the sequence of control strategies <`face-space-on-right`> *where* `face-space-on-right` *is a hill-climbing control strategy.*

[4] points `a` and `b` are not distinctive states.

At the causal level, the schemas $((V1, d1), A1, (V2, d2))$ *and* $((V2, d2), A2, (V3, d3))$ *are created, where* $A1$ *represents the sequence* `<get-into-corridor, follow-middle-line, hc-T-intersection>` *and* $A2$ *represents the sequence* `<face-space-on-right>`. *At the topological level,* $A1$ *is a* `Travel` *action while* $A2$ *is a* `Turn` *action.*

Next we describe the spatial representation that can be derived from a given set of schemas.

3.2 The SSH Causal Theory

The SSH Causal theory uses a sorted language with variables for *distinctive states, actions and views*. We use the following predicates in order to define the SSH causal theory.

1. *View(dp, V)*: V is the *view* at the *distinctive state* dp.
2. *CS(dp,A,dq)* : the *distinctive state* dq is a possible result of executing the *action* A at the *distinctive state* dp.[5]

Definition 1 *Consider a set S of schemas,* $S = \{s_1, \ldots, s_i = ((V_i, d_i), A_i, (V_i', d_i')), \ldots, s_n\}$.

1. *By definition, MAP(S) denotes the formula*

$$\bigwedge_{i=1}^{n} CS(d_i, A_i, d_i') \wedge View(d_i, V_i) \wedge View(d_i', V_i')$$

2. *Let* $\{V_{1'}, \ldots, V_{k'}\}$ *be the set of different view constant symbols mentioned in S. By definition, Views(S) denotes the formula*

$$UNA[V_{1'}, \ldots, V_{k'}]$$

That is, Views(S) is the uniqueness-of-names axiom for views in S.

We consider two possible (types of) actions: *turn and travel.* At the causal level, we only require that $turn \neq travel$. Further distinctions between these actions are defined at the topological level.[6]

The SSH causal theory associated with a set of schemas S, *CT(S)*, is given by the following circumscriptive theory. [7] [8] [9]

$$Map(S) \tag{1}$$
$$Views(S) \tag{2}$$
$$CS(dp, A, dq) \rightarrow dp \neq dq \tag{3}$$
$$\forall dp \exists! V\, View(dp, V) \tag{4}$$
$$turn \neq travel \tag{5}$$
$$\mathbf{circ}\, CS \succ View\, \mathbf{var}\, \vec{d} \tag{6}$$

where $\vec{d}$ is the vector of distinctive state constant symbols occuring in S. The extent of CS and View defined by CT(S) corresponds to the spatial representation at the causal level (i.e. the *causal map*).

[5] CS stands for Causal Schema

[6] In the full definition of the SSH we consider the parametric actions *turn (a)* and *travel (d)* where a and d are monotonically related to the amount of turning and traveling respectively. This extra information is considered at the metrical level which we do not consider here.

[7] Throughout this paper we assume that formulas are universally quantified

[8] the formula $\exists! V\, P(V)$ means *"there exists a unique V s.t. P(V)"*

[9] The symbol $\succ$ indicates prioritized circumscription (see [10] section 7.2). We define our theories following the notation in [10] section 4, where instead of writing CIRC[A;Ab;Z], we list the theory axioms, A, followed by the circumscription policy, **circ** Ab **var** Z.

The purpose of the circumscription policy (6) is to minimize the number of distinctive states by declaring as equal two distinctive states that cannot be distinguished through actions or views [10]. By prioritizing CS over View we guarantee that only the schemas in S are used to define the extent of CS.[11]

Suppose that views uniquely determine distinctive states (i.e. $View(dp, V) \wedge View(dq, V) \equiv dp = dq$). In this case, the causal theory $CT(S)$ is completely determined by $Map(S)$. However, the sensory capabilities of a realistic robot agent may not be sufficient to distinguish distinctive states. Nevertheless, the causal theory $CT(S)$ embodies the default assumption that views uniquely determine distinctive states. Example 2 demonstrates a case when this default fails, leading the agent to conclude that two spatially different but indistinguishable states are the same.

Example 2 *Consider the set of schemas S given by* $\{((V, dp), travel, (V, dq)), ((V, dq), travel, (V, dr))\}$ *which is obtained by the robot while navigating the environment in figure 2. From axiom (3) we can conclude that* $dp \neq dq$ *and* $dq \neq dr$*. Since the same view is experienced at dp,dq and dr, the extent of CS and View is minimized by declaring* $dp = dr$*.*

[10] That is why we have $\vec{d}$ as varied constants in (6): indistinguishable distinctive places must be the same in order to minimize the extent of CS and View

[11] This corresponds to a "schema completion": in order to minimize the set of distinctive states, the agent cannot use actions it did not execute. In addition, notice that constant symbols representing views are not varied constants in the circumscription policy : two views cannot be identified in order to minimize the extent of CS or View. (more later)

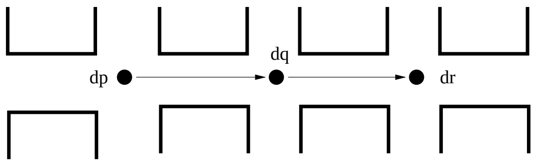

Figure 2: Distinctive states *dp* and *dr* cannot be distinguished at the causal level. Topological information is needed in order to distinguish them. (see text)

Though dp and dr were experienced at different states of the environment, at the causal level they are declared as equal. This happens because neither the actions nor the views provide enough information to distinguish them. In the next section we introduce the SSH topological theory. We will show that dp and dr can be distinguished by using topological information.

3.3 The SSH Topological Theory

In addition to the sorts and predicates used by the SSH causal theory, the SSH topological theory assumes a sorted language with variables for *paths and places* as well as the following predicates.

1. *dpath(p)* : the *path* p is a *directed* path
2. *at(dp,p)* : the *distinctive state* dp is at *place* p
3. *PO(p,pa,q)* : *place* p is before *place* q on *path* pa.[12]

Roughly speaking, a *place* corresponds to a set of distinctive states linked by turn with no travel actions. Similarly, a *path* corresponds to a set of distinctive states linked by travel with

[12] PO stands for Path Order

no turn actions.

The SSH topological theory associated with a set of schemas S, *TT(S)*, is given by the following circumscriptive theory. [13] [14]

- Causal level axioms (1)-(5).
- Every distinctive state is at a unique topological place.

$$\forall dp \exists ! p\, at(dp, p) \tag{7}$$

There is at least one distinctive state at every topological place.

$$\forall p \exists dp\, at(dp, p) \tag{8}$$

- PO(· ,pa,·) is a discrete linear order for every path pa.

$$PO(p, pa, q) \rightarrow \neg PO(q, pa, p) \tag{9}$$

$$PO(p, pa, q) \wedge PO(q, pa, r) \rightarrow PO(p, pa, r) \tag{10}$$

$$\begin{aligned} onpath(p, pa) \wedge onpath(q, pa) \wedge p \neq q \rightarrow \\ PO(p, pa, q) \vee PO(q, pa, p) \end{aligned} \tag{11}$$

- After turning, the robot is at the same place.

$$\begin{aligned} CS(dp, turn, dq) \wedge at(dp, p) \wedge at(dq, q) \\ \rightarrow p = q \end{aligned} \tag{12}$$

- After traveling, the agent is at a different place on the same directed path.

$$\begin{aligned} CS(dp, travel, dq) \wedge at(dp, p) \wedge at(dq, q) \rightarrow \\ p \neq q \wedge \exists pa\, dpath(pa) \wedge PO(p, pa, q) \end{aligned} \tag{13}$$

- The agent must have traveled between consecutive places on a path.

$$\begin{aligned} next(p, pa, q) \rightarrow \exists\, dp, dq\, \{at(dp, p) \\ \wedge at(dq, q) \wedge CS(dp, travel, dq)\} \end{aligned} \tag{14}$$

- Consecutive places on a path are linked by Travel actions.

$$\begin{aligned} next(p, pa, q) \wedge next(q, pa, r) \rightarrow \\ \exists\, dp, dq, dr\, \{at(dp, p) \wedge at(dq, q) \wedge at(dr, r) \wedge \\ CS(dp, travel, dq) \wedge CS(dq, travel, dr)\} \end{aligned} \tag{15}$$

- Circumscription policy.

$$\mathbf{circ}\, CS \succ dpath \succ PO \succ View\, \mathbf{var}\, \vec{d} \tag{16}$$

The extent of *dpath*, *PO* and *at*, as defined by TT(S), corresponds to the spatial representation at the topological level (i.e. the *topological map*). The circumscription policy (16) aims to identify the minimum number of directed paths while declaring as equal distinctive states that are not distinguishable by topological relations or views.

We prevent the agent from inferring the existence of schemas it did not actually experience (i.e. "hallucinations") by giving the highest priority to CS in (16). The next example shows how "phantom" schemas could be used in order to minimize the number of directed paths.

Example 3 *Consider the set of schemas indicated in the environment of figure 3. From (7,12,13) three topological places, A,B and C,*

[13] onpath(p,pa) is an abbreviation for $\exists r PO(p, pa, r) \vee PO(r, pa, p)$

[14] next(p,pa,q) is an abbreviation for $PO(p, pa, q) \wedge \forall r\, r \neq q \wedge PO(p, pa, r) \rightarrow PO(q, pa, r)$

must be created such that $A \neq B$, $B \neq C$, at(d1,A), at(d2,B), at(d3,B) and at(d4,C) are true. From (13), there exists at least two directed paths P1 and P2, such that PO(A,P1,B) and PO(B,P2,C) are true.

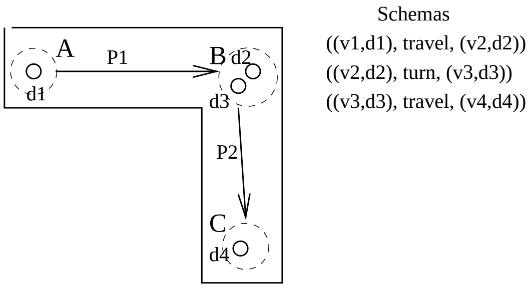

Figure 3: By inferring the existence of the "phantom" schema ((v2,d2), travel, (v4,d4)) the agent can conclude that paths P1 and P2 are the same, and consequently minimize the number of directed paths. Phantom schemas are ruled out by giving the highest priority to CS in (16).

Suppose that we give the highest priority to the predicate dpath in order to minimize the number of directed paths at the cost of changing the extent of the other predicates. In that case, a minimal model will have P1=P2 by including CS(d2,travel,d4)[15]*. Since this relation does not represent an actual schema experienced by the agent, we must enforce that only the schemas in S are used to construct the topological map. We achieve this goal by giving CS the highest priority in the circumscription policy (16).*

As the SSH topological level is built on top of the causal level, when using topological information the causal map corresponds to the extent of CS and View as defined by TT(S). *The next example shows how the causal map changes as topological considerations are included in the space representation. Topological information can dictate spatial distinctions that cannot be derived from causal information alone.*

Example 4 *Consider the same set of schemas in example 1. By (7) there exists topological places p,q and r such that at(dp,p), at(dq,q) and at(dr,r). Using axiom (13) we conclude that $p \neq q$ and $q \neq r$. We minimize the extent of dpath by postulating a directed path, pa, such that PO(p,pa,q) and PO(q,pa,r) are true. By (10) we conclude that PO(p,pa,r) and consequently, $p \neq r$. From the uniqueness condition in (7) it follows that $dp \neq dr$.*

Since the predicate PO conveys all the information required to define a network among the different places, it would seem that the use of the predicate *dpath* is not required. As the next example shows, minimizing the extent of PO does not guarantee a minimum set of paths. By using the predicate *dpath* we achieve this goal.

Example 5 *Consider the same set of schemas in example 2 and assume that we do not have the predicate dpath. There will be then two ways of minimizing the extent of PO.*

1. *Postulate the existence of a unique path pa, such that the extent of PO is given by PO(p,pa,q), PO(q,pa,r) and PO(p,pa,r). Axiom (9) implies that $p \neq r$.*
2. *Postulate the existence of two different paths p1 and p2 such that PO's extent is given by PO(p,p1,q) and PO(q,p2,r). Further minimizations will imply that p=r.*

[15] The need for this relation is derived from axiom (14), since P1=P2 would imply next(A,P1,B) and next(B,P1,C).

Models of the second option above do not agree with our intuition of what a path is and consequently should not be considered as possible representations of the topological map.

3.4 The effect of the agent's sensory capabilities on the topological map

While the SSH correctly defines the topology of most common environments, there is still pathological cases in which poor sensory capabilities or the symmetry of the environment makes it difficult for the agent to extract the environment's correct topology. The next discussion illustrates this point.

Let's assume that we have two agents, A and B, whose environment is a square room. Both agents visit the different corners of the room in the same order, as suggested by figure 4. Agent A's sensory apparatus allows it to define *views* by characterizing the direction of walls and open space. Accordingly, agent A experiences *four* different views, *V1-V4*, in this environment (see figure 5).

In addition to the sensory apparatus of agent A, agent B has a *compass* which allows it to experience *sixteen* different views in its environment, each from the set $\{(V_i, ori) : i = 1, \ldots, 4,\ ori \in \{N, E, S, W\}\}$.[16] Next we describe the topological map derived by agents B and A, respectively.

Agent B's topological Map. Notice that agent B's views uniquely determine the environment distinctive states but $d1$ and $d17$

[16]N,E,S and W stand for North, East, South and West respectively.

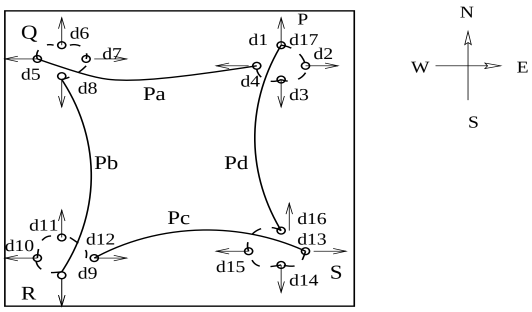

Figure 4: The figure shows the sequence of actions followed by the agents A and B while navigating a square room. Starting at distinctive state d1, distinctive states are visited in the order suggested by their number. Dashed lines indicate Turn actions. Solid lines indicate Travel actions. Distinctive states d1 and d17 are at the same physical place, but this information is not available to the agents.

View	Wall's direction	Open space direction
V1	ahead, right	behind, left
V2	ahead, left	behind, right
V3	left, behind	right, ahead
V4	right,behind	left, ahead

Figure 5: Definition of views V1-V4. Each view is characterized by the direction of walls and open space.

(see figure 6). The circumscription policy (16) will declare $d1 = d17$ and a model of TT(S) gives the expected topology associated with the square room (see figure 7).

Agent A's topological Map. As mentioned before, agent A only experiences four different views, *V1-V4*, in its environment. The set of distinctive states associated with each one of these views is as follows:[17] {d1,d5,d9,d13,d17}

[17]The set of distinctive states associated with a view V is the set of distinctive states that share the same view

View	Distinctive State	View	Distinctive State
(V1,N)	d1,d17	(V2,N)	d6
(V1,E)	d13	(V2,E)	d2
(V1,S)	d9	(V2,S)	d14
(V1,W)	d5	(V2,W)	d10
(V3,N)	d11	(V4,N)	d16
(V3,E)	d7	(V4,E)	d12
(V3,S)	d3	(V4,S)	d8
(V3,W)	d15	(V4,W)	d4

Figure 6: Agent B's associations between views and distinctive states. The set of distinctive states associated with each view is shown.

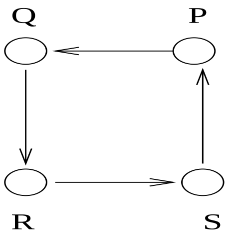

Figure 7: Environment's topology deduced by agent B.

is associated with V1, {d2,d6,d10,d14} is associated with V2, {d3,d7,d11,d15} is associated with V3, and {d4,d8,d12,d16} is associated with V4.

The circumscription policy (16) will imply that d1=d9, d2=d10, d3=d11, d4=d12, d5=d13, d6=d14, d7=d15, and d8=d16 , and consequently only two directed paths are needed in a minimal model of TT(S). The resulting topology is depicted in figure 8. The environment looks perfectly symmetric to the agent. In cases like this, information at the SSH metrical level is used to establish further spatial distinctions. Next we describe how Agent A can use this information for getting the square room topology.[18]

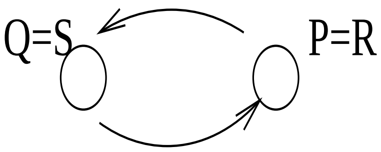

Figure 8: Environment's topology deduced by agent A.

Using the SSH metrical level. Let P, Q, R, and S be the topological places associated with *d4*, *d8*, *d12* and *d16* respectively. Let Pa be the directed path from P to Q, Pb be the directed path from Q to R, and Pc be the directed path from R to S (see figure 4). Metrical information derived from the turn actions is used to estimate the angle among the different distinctive states associated with place Q. We can conclude that the angle between *d5* and *d8* is about 90^o. This angle is then declared as the angle between Pa and Pb at place Q, denoted by $angle(Pa, Pb, Q, 90^o)$. If we are to suppose that directed paths represent "straight lines" in the environment, we can conclude then that $P \neq R$ and $\neg onpath(R, Pa)$. Since $onpath(R, Pc)$ is true, it follows that $Pa \neq Pc$. By a similar argument, it is the case that $angle(Pb, Pc, R, 90^o)$ and consequently Pa and Pc are "parallel paths". It follows then that $P \neq S$.

As illustrated in the previous discussion, the use of metrical information allows us to define further topological relations among places, which

V.

[18] We do not formalize the SSH metrical level in this paper. However, the example gives a glance of how this information can be used.

in turn translate into further distinctions among distinctive states.

4 Related work

The causal graph as defined by CT(S) can be seen as the minimum automaton that is consistent with the information in S. Researchers in the automata learning community [4, 5] have developed promising real time algorithms to solve this problem. However, since topological considerations determine the structure of the causal graph, these algorithms must be adapted to take these considerations into account.[19]

Work by Shanahan [6, 7] has recently addressed the problem of defining a logical account of sensor data assimilation. In this work, data assimilation is seen as an abduction task with respect to a given theory of action, change, space, shape, and the robot relationships to the world (i.e. the effect of the robot's actions on the world, and the effect the world has on the robot's sensors). We have interpreted the results of Shanahan's work as a possible formalization for some of the aspects involved in the transition from the SSH sensorimotor level to the SSH causal level.

5 Conclusions and Future work

The Spatial Semantic Hierarchy (SSH) is an ontological hierarchy of representations for knowledge of large-scale space. In large-scale space the structure of the environment is revealed by integrating local observations over time, rather than being perceived from a single vantage point. Accordingly, the robot's continuous interaction with the environment is summarized by schemas at the SSH causal level. These local schemas are then used to derive the global structure of the environment at the SSH topological level. The global structure of the environment is represented by a network (graph) of places arranged on paths showing the connectivity relations among them.

In this paper we have defined the SSH causal and topological theories. These theories formalize our intuition of some ideas informally described and used in physical robot implementations of the SSH. Having a logical account for the SSH causal and topological levels supplies us with a tool for clarifying (defining) some of the main aspects of the SSH. In particular, it will allow us to prove correct our current implementation of the SSH. More important, the SSH formalization presented in this paper shows how the different levels of the hierarchy are combined to determine spatial distinctions that can not be inferred by each SSH level alone.

We are still working on a formalization of the SSH metrical level based on qualitative reasoning methods. The complete formalization of the SSH will allow us to determine and justify the cognitive capabilites that are needed by an agent in order to capture the structure of its environment. In the same vein, the SSH formalization is a tool for studying the "grounding problem": how logical terms come to designate things.

[19] For example, the minimal automaton must satisfy the following constraint: the starting and resulting states after two consecutive travels must be different.

References

[1] Kuipers B. and Byun Y. T. A robust

qualitative method for spatial learning in unknown environments. In Morgan Kaufmann, editor, *AAAI-88*, 1988.

[2] Lee W.Y. Kuipers B., Froom R and Pierce D. The semantic hierarchy in robot learning. In J. Connell and S. Mahadevan, editors, *Robot Learning*, pages 141–170. Kulwer Academic Publishers, 1993.

[3] Kuipers B. A hierarchy of qualitative representations for space. In *Working papers of the Tenth International Workshop on Qualitative Reasoning about Physical Systems (QR-96)*. AAAI Press, 1996.

[4] Rivest R.L. and Schapire R.E. Inference of finite automata using homing sequences. *Information and Computation*, 103(2):299–347, April 1997.

[5] Basye K. Dean T. and Kaelbing L. Uncertainty in graph-based map learning. In Jonathan H. Connell and Sridhar Mahadevan, editors, *Robot Learning*, pages 171–192. Kluwer Academic Publishers, 1993.

[6] Shanahan M. P. Noise and the common sense informatic situation for a mobile robot. In *AAAI-96*, 1996.

[7] Shanahan M. P. Noise, non-determinism and spatial uncertainty. In *AAAI-97*, 1997.

[8] Kuipers B. and Byun Y. T. A robot exploration and mapping strategy based on semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.

[9] Kuo B. C. *Automatic Control Systems*. Prentice-Hall, Inc., fifth edition, 1987.

[10] Lifschitz V. Circumscription. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 297–352. Oxford University Press, 1994.