Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Artificial Intelligence 152 (2004) 47–104

Artificial Intelligence

www.elsevier.com/locate/artint

Towards a general theory of topological maps

Emilio Remolina [a,*], Benjamin Kuipers [b,1]

[a] *Stottler Henke Associates, Inc., 951 Mariners Island Blvd. #360, San Mateo, CA 94404, USA*
[b] *Computer Science Department, The University of Texas at Austin, Austin, TX 78712, USA*

Received 25 March 2002

Abstract

We present a general theory of topological maps whereby sensory input, topological and local metrical information are combined to define the topological maps explaining such information. Topological maps correspond to the minimal models of an axiomatic theory describing the relationships between the different sources of information explained by a map. We use a circumscriptive theory to specify the minimal models associated with this representation.

The theory here proposed is independent of the exploration strategy the agent follows when building a map. We provide an algorithm to calculate the models of the theory. This algorithm supports different exploration strategies and facilitates map disambiguation when perceptual aliasing arises.

Keywords: Topological maps; Causal maps; View graph; Spatial representations; Map building; Cognitive robotics; Spatial semantic hierarchy (SSH); Nested abnormality theories (NATs)

1. Introduction

Topological maps are graph-like spatial representations. Nodes in such a graph often represent states in the agent's configuration space and edges represent system trajectories that take the agent from one state to another. A hierarchical structure can be accommodated on top of this "behavior graph", where nodes at one level of the hierarchy represent

* Corresponding author.
E-mail addresses: eremolin@stottlerhenke.com (E. Remolina), kuipers@cs.utexas.edu (B. Kuipers).
[1] This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics Lab is supported in part by the Texas Higher Education Coordinating Board, Advanced Technology Program (grant 003658-0656-2001), NSF grants IRI-9504138 and CDA 9617327, by funding from Tivoli Corporation, and by an IBM Faculty Research Award.

0004-3702/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0004-3702(03)00114-0

sets of nodes in lower levels. Despite their common use, there is no consensus about what topological maps are, or how they are built. The meanings of nodes and edges in a topological map varies according to the application as well as the algorithms used to build them. Richer structures than the graph-like description above are sometimes adopted as part of what a topological map is. Nevertheless, there are common elements to most of the topological map descriptions, namely, the use of sensory input descriptions in order to identify nodes, connectivity relations among nodes in the map, and local metrical information associated with edges in the map.

In this paper, we present a general theory of topological maps whereby sensory input, topological and local metrical information are combined to define the topological maps explaining such information. We take a declarative approach to define what topological maps are and how they are related to the information used to build them. We distinguish between the *causal graph*, which is a transition graph representation of regularities in action and sensory experience, and the *topological map*, which represents spatial properties of actions and of places and paths in the environment. We define topological maps as the minimal models of an axiomatic theory describing the relationships between the different sources of information explained by a map. We provide an algorithm to calculate the models of the theory. This algorithm supports different exploration strategies and facilitates map disambiguation when perceptual aliasing arises.

The major assumption underlying the topological approach to mapping is that there is a level of abstraction of the underlying environment at which actions are deterministic. In the Spatial Semantic Hierarchy [31], this is achieved by defining *distinctive states* and *actions* composed of trajectory-following and hill-climbing control laws such that actions are functionally deterministic when applied between distinctive states (see Fig. 1). There are two other assumptions that, when true, allow us to state the axiomatic theory in simpler terms. These are the assumptions that (a) a path does not intersect itself, and (b) a distinctive state corresponds to at most one path and one direction on that path. Section 5.3 describes the more elaborate default theory required to handle environments that violate these assumptions.

2. Related work

Causal and topological maps have been mainly studied by cognitive theories of space and robotics. Cognitive theories of space are interested in the cognitive map, the human knowledge of large-scale space. Robotics is interested in representations of space that can be used (and learned) by an autonomous robot.

Computational theories of the cognitive map have been proposed in [9,21,23,30,34,40,44]. These theories account for incomplete knowledge of space, use of multiple frames of reference, qualitative representation of metrical information, and connectivity relations among landmarks. The theories differ on how sensory information is represented, what a place is, and how the overall spatial knowledge is structured.

The use of topological maps in robotics varies according to the type of information used when building such map. The sequence of views and actions generated by the robot exploration to recover the minimum deterministic automaton consistent with such information is used in [2,10,11,50]. In these works, actions do not have any spatial properties associated with them. Metrical information associated with actions is considered in [22,53], but there sensory information (views) is not used. The use of both sensory and metrical information is proposed in [9,16,25,55]. Among these works, [9,25] propose the use of multiple metrical frames of reference: the places in the topological map are not necessarily embedded in a single two dimensional Euclidean frame of reference, nor it is necessary to do so in order to create the topological map. In [28,31] the existence of topological objects (i.e., paths, regions) is proposed that can explain the agent's experiences without relying on metrical information but rather qualitative spatial properties (i.e., travel, turnRight, turnLeft, turnAround) associated with actions.

In research on physical robots by Lee [33] and Choset and Nagatani [7], effort has been put on describing how the agent solves the problem of "perceptual aliasing" (i.e., different places that share the same view). Different exploration strategies as well as different discrimination procedures are proposed to solve this problem. The description of topological maps is usually closely tied to the algorithms and exploration strategy used by the agent. It is difficult then to know what topological maps are and how they are related to the agent's experiences. The work by Choset and Nagatani [7] exploits the topology of the robot's free space to localize the robot on a partially constructed map. The map used in this work is the generalized Voronoi graph (GVG) which is a topological map that also encodes some metric information about the robot's environment. Our definition of topological maps includes but is not limited to GVGs. We propose an axiomatic theory of topological maps. The task of building the map is stated as an abduction task [47,52] where the agent's map correspond to the minimal models among those that explains its observations. Stating the minimality conditions as well as the ontology of the spatial representation is the content of this paper.

Metrical grid-based maps are another spatial representation used in the robotics community [4,15,59]. In these approaches the location of objects in a two dimensional Euclidean space are used to explain the agent's experiences. Topological maps as described in this paper can use metrical maps but they are confined to places, paths and local two dimensional frames of reference associated with regions.

The Spatial Semantic Hierarchy [31] assumes that an agent first builds a network of places and paths on top of which metrical models are added, rather than to build first a single metrical map from which a network of places and paths is derived. This assumption is motivated by research on human cognitive maps [37,46,54]. For the engineering tasks of robot exploration, mapping, and navigation, we believe that the "topology-first" approach is more efficient and robust. For example, Thrun et al. [59] propose a method for integrating topological and metrical paradigms to solve the concurrent mapping and localization problem studied in the mobile robotics community. The method has two phases. In the first phase, the topological mapping solves a global position alignment problem between potentially indistinguishable, significant places. The subsequent metric mapping phase produces a fine-grained metric map of the environment in high resolution. "This work illustrates that topological approaches indeed scale up to large and highly ambiguous environments. The environments tested here are difficult in that they possess large cycles, and in that local sensor information is insufficient to disambiguate locations" [59].

Finally, there are also feature-based spatial representations [58] where the map is a graph whose nodes represent observed features and whose edges represent geometric relationships between these features. Under these approaches the locations of geometric features in the environment and the position of the vehicle is jointly estimated in a stochastic framework. Like grid-based methods, feature-based methods are subject to cumulative metrical error and the difficulty of properly closing large loops. A major benefit of topological maps is that the problem of correctly closing large loops is separated from the problem of metrical mapping of local environments. We refer the reader to Borenstein's book [5, Chapter 8] for a review of different approaches to map building.

This article is organized as follows: in Section 3 we define how the agent represents its experiences in the environment. Section 4 defines the causal map representation. The topological theory is presented in three parts: Section 5 introduces the main properties of paths and places. Section 6 adds boundary relations to this representation, and Section 7 defines the use of local metrical information. Section 8 presents our algorithms to build the topological maps associated with the agent's experiences. Finally, we present our conclusions in Section 9.

3. The agent's experiences in the environment

We assume that the continuous interaction of the agent and its environment is summarized by a discrete *view-action-view* sequence of the form

$$v_0, a_0, v_1, a_1, \ldots, a_{n-1}, v_n. \tag{1}$$

A **view** represents a sensory description associated with an environment state. Only the name and not the internal structure of a view matters. The environment states where the views in sequence (1) were observed are called **distinctive states** (dstates). Note that distinctive states represent not only location, but also the agent's orientation in the environment. The same view can occur at different distinctive states (*perceptual aliasing*). It is possible for the agent to associate different distinctive state names with the same environment state. This is the case since the agent might not know at which of several environment states it is currently located. It is the purpose of the causal and topological theories (Sections 4 and 5) to deduce which of these dstates names refer to the same environment state.

An **action** denotes a sequence of one or more control laws [32] that take the agent from one dstate to the next. For example, in [25,26,31] distinctive states are the result of following *trajectory-following* and then *hill-climbing* control laws. The basin of attraction of the hill-climbing control laws absorbs accumulated error from each trajectory-following control law, along each action. Even with realistic levels of accuracy in the control laws, if the initial basin of attraction is large enough, and the hill-climbing control law is effective enough, the action become functionally deterministic (Fig. 1).

The sequence (1) is transformed into a set of **schemas** of the form $\langle (v_i, ds_i), a_i, (v_{i+1}, ds_{i+1}) \rangle$, where ds_i is the dstate name associated with the environment state where view v_i is observed. A schema represents a particular action execution of the agent in the environment. An action execution is characterized in terms of the distinctive states the agent was at before and after the action was performed.

Example 1. Consider the environment in Fig. 2. In order to go from distinctive state $ds1$ to distinctive state $ds2$, the agent executes the sequence of control laws

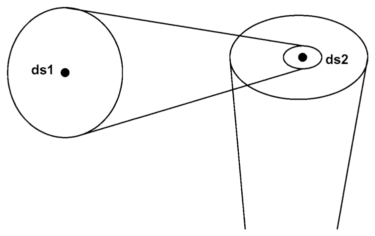

Fig. 1. Actions between distinctive states are functionally deterministic. The control laws making up an action have a basin of attraction surrounding the initial distinctive state ($ds1$). Any trajectory starting in that basin moves toward the fixed-point of the hill-climbing control law. Since any implementation has finite precision, the action terminates in a small region around the destination distinctive state ($ds2$). As long as the final region is small enough to be contained within the initial basin of attraction of every subsequent action departing from that state, then actions are functionally deterministic.

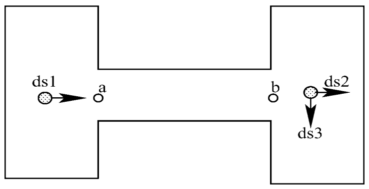

Fig. 2. A sequence of control strategies, $\langle$*get_into_corridor*, *follow_middle_line*, *localize*$\rangle$, takes the agent from distinctive state *ds*1 to distinctive state *ds*2. This continuous motion is represented by the schema $\langle (v1, ds1), a1, (v2, ds2)\rangle$, where $v1$ and $v2$ are the views at *ds*1 and *ds*2, and the action symbol $a1$ represents the sequence of control laws.

$\langle$*get_into_corridor*, *follow_middle_line*, *localize*$\rangle$ where *get_into_corridor* is a trajectory following control law that moves the agent from *ds*1 to a, *follow_middle_line* is a trajectory following control law that takes the agent from a to b, and *localize* is a hill-climbing control law that takes the agent from b to the distinctive state *ds*2. Environment states a and b are not distinctive states. At the distinctive state *ds*2 the agent is facing the wall ahead and it is equidistant from this wall and the intersection corners.

Distinctive state *ds*3 is at the same physical location as *ds*2 but with a different orientation. When the robot is at *ds*3, it is facing the open space (corridor) to the right of *ds*2. In order to go from distinctive state *ds*2 to distinctive state *ds*3, the agent executes the sequence of control strategies $\langle$*face_space_on_right*, *localize*$\rangle$. The schemas $\langle (v1, ds1), a1, (v2, ds2)\rangle$ and $\langle (v2, ds2), a2, (v3, ds3)\rangle$ are created, where $a1$ and $a2$ are action symbols representing the respective sequence of control laws.

4. Causal graphs

Schemas summarize the continuous interactions of the agent in the environment. This is done by storing the initial and final distinctive states (and their corresponding views) for any action execution. By considering only the views associated with the initial and final distinctive states of a schema, we define the *view graph* (Section 4.2.1), which relates different views by actions linking them. By considering sequences of actions as well as views, the agent can further distinguish distinctive states. In Section 4.3 we define the predicate ceq which is the case for distinctive states that are not distinguishable by actions and views. We then define the *causal graph* whose nodes are classes of distinctive states (classes w.r.t. ceq). This representation is akin to the view graph although it imposes further refinement in the set of environment states that are consistent with the agent experiences.

4.1. Ontology of the causal theory

We use a first order sorted language in order to describe causal graphs. The sorts of such language include *views*, *actions*, *action types*, *action qualitative descriptions*, *distinctive states* and *schemas*. Next we present the predicate symbols and axioms associated with this ontology.

We use the predicate **View(ds, v)** to represent the fact that v is the *view* associated with *distinctive state* ds. We assume that a distinctive state has a unique view,[2,3]

$$\exists! v \; View(ds, v). \tag{2}$$

However, we do **not** assume that views uniquely determine distinctive states (i.e., $View(ds, v) \wedge View(ds', v) \not\rightarrow ds = ds'$). This is the case since the sensory capabilities of an agent may not be sufficient to distinguish distinctive states.

An action has a unique type, either *travel* or *turn*, associated with it.[4] These constant symbols define completely the sort of *action_types* (Axiom 3). The predicate **Action_type(a, type)** represents the fact that the type of action a is *type*. Formally,

$$turn \neq travel, \; \forall atype \; \{atype = turn \vee atype = travel\}, \tag{3}$$

$$\exists! type \; Action\_type(a, type). \tag{4}$$

Turn actions have associated a unique qualitative description. The sort of qualitative descriptions is completely defined by the constant symbols *turnLeft*, *turnRight* and *turnAround* (Axioms 5 and 6). We use the predicate **Turn_desc(a, desc)** to indicate that *desc* is the qualitative description of the *turn action a*. Formally,[5]

$$UNA[turnLeft, turnRight, turnAround], \tag{5}$$

$$\forall desc \; \{desc = turnLeft \vee desc = turnRight \vee desc = turnAround\}, \tag{6}$$

$$Turn\_desc(a, desc) \rightarrow Action\_type(a, turn), \tag{7}$$

$$Action\_type(a, turn) \rightarrow \exists! desc \; Turn\_desc(a, desc). \tag{8}$$

A **schema** represents a particular action execution of the agent in the environment. We use the following predicates to represent information associated with a schema: **action(s, a)**,

[2] Throughout this paper we assume that free variables in formulas are universally quantified.

[3] The formula $\exists! v \; P(v)$ means *“there exists a unique v s.t. $P(v)$”*. Formally, $\exists v \forall x \, [P(x) \equiv x = v]$.

[4] The type of an action will be important in the topological theory (Section 5). For completeness of the presentation we introduce this concept here.

[5] The notation $UNA[t_1, \ldots, t_n]$ represents the uniqueness of names axioms for the grounded terms $t_1, \ldots, t_n$. These axioms require that $t_i \neq t_j$ for $i \neq j$.

action a is the action associated with *schema s*; **context(s, ds)**, *ds* is the starting *distinctive state* associated with the action execution represented by *schema s*; and **result(s, ds)**, *ds* is the ending *distinctive state* associated with the action execution represented by *schema s*. While we require a unique context and action associated with a schema, the result of a schema is optional (but unique if it exists):

$$\exists! a\, action(s, a),\ \exists! ds\, context(s, ds),\ result(s, ds) \wedge result(s, ds') \rightarrow ds = ds'. \tag{9}$$

Most often we are interested in *complete* schemas: those for whom the resulting distinctive state exists. Nevertheless, incomplete schemas allow the representation to account for common states of incomplete knowledge like "I could take you there, but I cannot tell you how" [31]. We use the (Causal Schema) predicate $CS(s, ds, a, ds')$ defined as

$$CS(s, ds, a, ds') \equiv_{def} context(s, ds) \wedge action(s, a) \wedge result(s, ds') \tag{10}$$

to express the fact that schema *s* represents an execution of action *a* which took the agent from *distinctive state ds* to distinctive state *ds'*.

An action execution also has metrical information associated with it. This metrical information represents an estimate of, for example, the distance or the angle between the distinctive states associated with the action execution. We defer the study of metrical information associated with schemas until Section 7.

While schemas are explicit objects of our theory, it is convenient to leave them implicit. We introduce the following convenient notation:[6]

$$\begin{aligned}
&\langle ds, a, ds'\rangle \equiv_{def} \exists s\, CS(s, ds, a, ds'),\\
&\langle v, a, v'\rangle \equiv_{def} \exists s, ds, ds' \left\{CS(s, ds, a, ds') \wedge View(ds, v) \wedge View(ds', v')\right\},\\
&\langle (v, ds), a, (v', ds')\rangle \equiv_{def} \exists s \left\{CS(s, ds, a, ds') \wedge View(ds, v) \wedge View(ds', v')\right\},\\
&\langle ds, type, ds'\rangle \equiv_{def} \exists s, a \left\{CS(s, ds, a, ds') \wedge Action\_type(a, type)\right\},\\
&\langle ds, desc, ds'\rangle \equiv_{def} \exists s, a \left\{CS(s, ds, a, ds') \wedge Turn\_desc(a, desc)\right\}.
\end{aligned}$$

4.2. The E formulae

The agent's experiences in the environment, *E*, are described in terms of *CS, View, Action_type* and *Turn_desc* formulae. Associated with *E* we have the sets *S(E), DS(E), V(E), A(E)* of schemas, distinctive states, views and action constant symbols occurring in *E*. We require all these symbols to be different (i.e., *uniqueness of names* assumption) and to completely define their corresponding sorts (*domain closure* assumption):

$$\begin{array}{llll}
UNA[s_1, \ldots, s_k], & s_i \in S(E), & UNA[ds_1, \ldots, ds_l], & ds_i \in DS(E),\\
UNA[a_1, \ldots, a_n], & a_i \in A(E), & UNA[v_1, \ldots, v_m], & v_i \in V(E),
\end{array} \tag{11}$$

$$\begin{array}{ll}
\forall s \bigvee_{s_i \in S(E)} s = s_i, & \forall ds \bigvee_{ds_i \in DS(E)} ds = ds_i,\\
\forall a \bigvee_{a_i \in A(E)} a = a_i, & \forall v \bigvee_{v_i \in V(E)} v = v_i.
\end{array}$$

[6] Notice that we have "overloaded" the bracket notation depending on the type of its arguments.

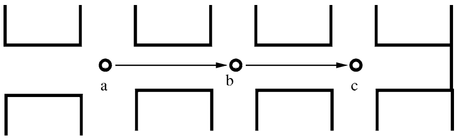

Fig. 3. The agent moves among corridor intersections that have the same view $v+$. a, b and c are the distinctive states where this view is observed at.

The axioms above are not only required from a logical point of view, but make sense from the knowledge representation point of view. Domain closure axioms prevent models from including objects different from those experienced (named) by the agent. Each of the agent schemas represents a different experience and the agent names them with a different schema constant symbol. Different view symbols represent different sensory input. This is the case since the agent decides what view to associate with a sensory input. *Different distinctive state constant symbols might represent the same environment state.* Nevertheless, we assume that different distinctive state symbols are interpreted by different elements of the sort of distinctive states and we use the predicate *ceq* (Causally Equal) to indicate whether two distinctive states represent the same environment state (Section 4.3).

Finally, the type of actions as well as the qualitative description of turn actions have to be specified as part of the formulae E:

$$\mathit{Action\_type}(a, \mathit{type}) \equiv \bigvee_{\mathit{Action\_type}(a_i, \mathit{type}_i) \in E} [a = a_i \wedge \mathit{type} = \mathit{type}_i], \tag{12}$$

$$\mathit{Turn\_desc}(a, \mathit{desc}) \equiv \bigvee_{\mathit{Turn\_desc}(a_i, \mathit{desc}_i) \in E} [a = a_i \wedge \mathit{desc} = \mathit{desc}_i]. \tag{13}$$

Definition 1. Given a set E of *CS*, *View*, *Action_type* and *Turn_type* formulae,

$$\mathit{COMPLETION}(E)$$

denotes the union of E with Axioms (11)–(13).

Example 2. Consider the set of experiences E gathered by the agent while navigating the environment in Fig. 3. The agent moves among intersections by performing action *ml*. The sensory input at the different intersections is very similar, and the agent associates the view $v+$[7] with the different distinctive states it found (i.e., a, b and c).

The elements of E are as follows: $\mathit{Action\_type}(ml, \mathit{travel})$, $CS(s1, a, ml, b)$, $CS(s2, b, ml, c)$, $\mathit{View}(a, v+)$, $\mathit{View}(b, v+)$, and $\mathit{View}(c, v+)$.

The uniqueness of names axioms associated with E are $s1 \neq s2$ and $a \neq b \wedge a \neq c \wedge b \neq c$. The domain closure axioms associated with E are $\forall s\, \{s = s1 \vee s = s2\}$, $\forall ds\, \{ds = a \vee ds = b \vee ds = c\}$, $\forall a\, \{a = ml\}$ and $\forall v\, \{v = v+\}$.

Finally, we also have the axioms $\forall a, \mathit{desc}\, \{\mathit{Turn\_desc}(a, \mathit{desc}) \equiv \mathit{false}\}$ and $\forall a, \mathit{type}\, \{\mathit{Action\_type}(a, \mathit{type}) \equiv [a = ml \wedge \mathit{type} = \mathit{travel}]\}$.

[7] As with any other symbol name, the view name is arbitrary. The + in the view name is used to indicate that the view corresponds to a four corridor intersection. Later we use the symbol $\sqsupset$ to indicate that the view corresponds to an end of corridor.

4.2.1. The view graph

The *view graph* associated with a set of experiences E is the labeled graph $\langle \mathit{Nodes}, \mathit{Edges}, \mathit{Labels}\rangle$ such that:

- $\mathit{Nodes} = V(E)$, $\mathit{Labels} = A(E)$.
- $\mathit{Edges} = \{(v, a, v')\colon \mathit{COMPLETION}(E) \models \langle v, a, v'\rangle\}$.

When the same view occurs at different environment states, the view graph is not very informative. The agent has to use information other than the views alone in order to distinguish different environment states (see next section and Section 5). However, should the agent have enough sensory capabilities as to distinguish distinctive states by their views, then the view graph becomes a powerful spatial representation for reliable navigation. Work in [17,38,51,57] shows how the view graph is consistent with human navigation abilities.

4.3. The causal theory

We use the predicate $\mathbf{ceq}(\mathbf{ds}, \mathbf{ds}')$ to denote the fact that *distinctive states ds* and ds' are *causally* indistinguishable. (In Section 5 we define when distinctive states are topologically indistinguishable.) Informally, $\mathit{ceq}(ds, ds')$ is the case whenever distinctive states *ds* and ds' are indistinguishable by the actions and views in a given set of experiences E. The theory $\mathit{CT}(E)$ below defines the extent of the predicate *ceq*.

The causal theory associated with a set of experiences E, $\mathit{CT}(E)$, is the following nested abnormality theory (NATs) [36] (see Appendix A):

$$\mathit{CT}(E) = \tag{14}$$

$\mathit{COMPLETION}(E)$,

Axioms (2)–(10),

$$\langle ds, a, ds'\rangle \wedge \langle ds, a, ds''\rangle \rightarrow ds' = ds'', \tag{15}$$

$$\mathit{CEQ\_block} = \tag{16}$$

$\{$ *max ceq*:

$\mathit{ceq}(ds_1, ds_1)$,

$\mathit{ceq}(ds_1, ds_2) \rightarrow \mathit{ceq}(ds_2, ds_1)$,

$\mathit{ceq}(ds_1, ds_2) \wedge \mathit{ceq}(ds_2, ds_3) \rightarrow \mathit{ceq}(ds_1, ds_3)$,

$$\mathit{ceq}(ds_1, ds_2) \rightarrow \mathit{View}(ds_1, v) \equiv \mathit{View}(ds_2, v), \tag{17}$$

$$\mathit{ceq}(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \rightarrow \mathit{ceq}(ds_1', ds_2') \tag{18}$$

$\}$

Axiom (15) states our assumption that actions are deterministic. Axiom (17) states that indistinguishable distinctive states have the same view. Axiom (18) states that if distinctive states *ds* and ds' are indistinguishable, and action a is performed for both *ds* and ds', then the resulting distinctive states must also be indistinguishable. Axioms (17) and (18) allow us to prove that if ds and ds' are two indistinguishable distinctive states, then any sequence of actions executed at ds and ds' will render the same sequence of views.

Given an action symbol A and distinctive state ds, $A(ds) = ds'$ if the schema $\langle ds, A, ds' \rangle$ has been observed, otherwise, $A(ds) = \perp$. Moreover, $A(\perp) = \perp$. The definition is then extended to action sequences in the standard way. Notice that $A(ds)$ is well-defined given our assumption that actions are deterministic (Axiom (15)).

Lemma 1. *Let A denote a sequence of action symbols. Let $A(ds)$ denote the distinctive state symbol resulting from executing the sequence A starting at distinctive state ds, or $\perp$ if A is not defined for ds. Then,*

$$ceq(ds_1, ds_2) \wedge A(ds_1) \neq \perp \wedge A(ds_2) \neq \perp \rightarrow View(A(ds_1), v) \equiv View(A(ds_2), v).$$

There is a special case in which *ceq* is an equivalence relation without explicitly stating the axioms requiring so. This is the case when the result of every action at every distinctive state is known.

Definition 2. A set of experiences E is **complete** whenever

$$E \models \forall a, ds \exists ds' \langle ds, a, ds' \rangle.$$

Theorem 1. *Let E be a complete set of experiences and let CEQ_block be defined as follows*:

$$\begin{aligned}
&\{\ \mathit{max}\ ceq: \\
&\quad ceq(ds_1, ds_2) \rightarrow View(ds_1, v) \equiv View(ds_2, v), \\
&\quad ceq(ds_1, ds_2) \wedge \langle ds_1, a, ds'_1 \rangle \wedge \langle ds_2, a, ds'_2 \rangle \rightarrow ceq(ds'_1, ds'_2) \\
&\}
\end{aligned}$$

Then, the predicate ceq is an equivalence relation.

Proof. See Appendix B. □

When a set of experiences is complete the predicate *ceq* captures the idea that two distinctive states are the same if they render the same views under any sequence of actions.

Theorem 2. *Let E be a complete set of experiences. Then,*

$$ceq(ds_1, ds_2) \equiv \forall A, v \left[View(A(ds_1), v) \equiv View(A(ds_2), v)\right].$$

Proof. See Appendix B. □

Example 3. Consider the set of experiences E as in Example 2 (see Fig. 4(a)). Since the same view is experienced at a, b and c, the extent of *ceq* is maximized by declaring

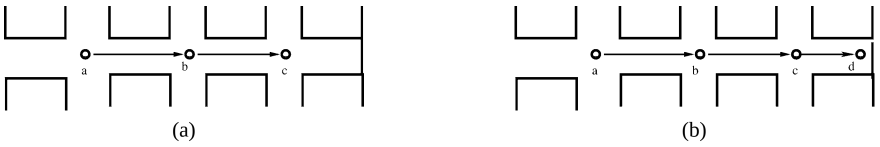

Fig. 4. (a) Distinctive states *a, b* and *c* cannot be causally distinguished. Topological information is needed in order to distinguish them (see text) (b) *a, b* and *c* are distinguished given the new information $\langle c, \mathit{travel}, d\rangle$.

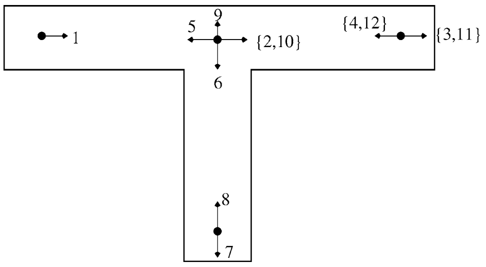

Fig. 5. The agent visits the different distinctive states in the order suggested by their numbers. The same view occurs at the different corners (i.e., $\mathit{view}(1) = \mathit{view}(4) = \mathit{view}\ (8)$). Three different causal models can be associated with the agent exploration of this T-environment (see text).

$\mathit{ceq} = \mathit{true}$ (i.e., $\forall x, y\ \mathit{ceq}(x, y)$). Notice that axiom (18) is trivially satisfied since no action has been executed at c.

Although *a, b* and *c* were experienced at different environment states, they are declared causally indistinguishable. This happens because neither the actions nor the views in E provide enough information to distinguish them. By using topological information (i.e., the concepts of *path* and *place*, see Section 5) we will be able to distinguish these distinctive states (see Example 5).

Suppose the agent continues exploring the environment and gets the new information $\mathit{View}(d, v\ \sqsupset)$, $\mathit{CS}(s3, c, ml, d)$, as suggested in Fig. 4(b). In virtue of Lemma 1, it can be seen that $\mathit{ceq}(ds, ds') \equiv ds = ds'$, and consequently the agent concludes that all distinctive states refer to different environment states.

Different models of $CT(E)$ generally arise when the set of experiences E is incomplete (i.e., the agent has not completely explored the environment) or when weak sensors determine the same view at different environment states.

Example 4. Consider the environment depicted in Fig. 5. The agent visits the different distinctive states as suggested by their numbers in the figure. The same travel action ml is performed when traveling from a corner to the intersection (i.e., $\langle 1, ml, 2\rangle$) and vice versa (e.g., $\langle 4, ml, 5\rangle$). A turn around action is performed when reaching a corner (e.g., $\langle 3, \mathit{change\_path\_direction}, 4\rangle$, $\langle 7, \mathit{change\_path\_direction}, 8\rangle$, etc.). Assume that the different corners have the same views (i.e., $\mathit{view}(1) = \mathit{view}(4) = \mathit{view}(8)$, $\mathit{view}(3) = \mathit{view}(7) = \mathit{view}(11)$), and views associated with the other distinctive states are different.

Three models of $CT(E)$ can be associated with the exploration E of the T-environment:

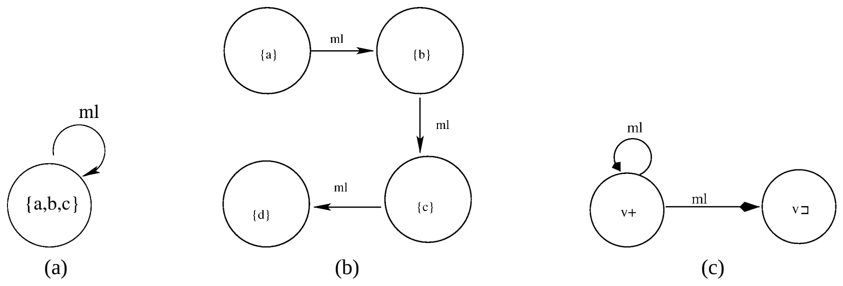

Fig. 6. (a)–(b). Causal graphs associated with the set of experiences in Figs. 4(a) and 4(b). (c) View graph associated with the set of experiences in Fig. 4(b). Notice that the causal and view graphs associated with the experiences in Fig. 4(a) are isomorphic.

1. Model 1: $ceq(8, 12), ceq(12, 8), ceq(x, x)$.[8]
2. Model 2: $ceq(1, 12), ceq(12, 1), ceq(x, x)$.
3. Model 3: $ceq(4, 12), ceq(12, 4), ceq(3, 11), ceq(11, 3), ceq(2, 10), ceq(10, 2), ceq(x, x)$.

In all the models above, $\neg ceq(1, 4)$, $\neg ceq(1, 8)$, $\neg ceq(4, 8)$. For instance, from $\langle 1, ml, 2\rangle$, $\langle 4, ml, 5\rangle$, and $view(2) \neq view(5)$ we conclude that $\neg ceq(1, 4)$. Although dstate 12 is at the same environment state as dstate 4, it is possible that $ceq(1, 12)$ or $ceq(8, 12)$. This is the case since no action has been performed at dstate 12.

Notice that the models of $CT(E)$ are maximal with respect to the set inclusion for ceq. The number of elements in the possible extents of ceq could vary, and consequently the number of different environment states represented by the models of $CT(E)$ will also vary. For instance, the three models above represent 11, 11 and 9 environment states respectively.

Finally, notice that all the models above are possible since at the causal level turn and travel actions do not bear any spatial meaning. When we consider topological information, only model 3 above will be possible (see Example 10).

4.4. The causal graph

The causal graph associated with a set of experiences E is the labeled graph $\langle Nodes, Edges, Labels\rangle$ such that:

- $Nodes = DS(E)/ceq$, $Labels = A(E)$,
- $Edges = \{([ds], a, [ds]')\colon\ COMPLETION(E) \models \langle ds, a, ds'\rangle\}$

where $DS(E)/ceq$ denotes the set of equivalence classes of $DS(E)$ modulo ceq, and $[ds]$ denotes the equivalence class of ds given ceq.

The problem of distinguishing environment states by outputs (views) and inputs (actions) has been studied in the framework of automata theory [1,2,20,50]. In this framework, the problem we address is the one of finding the minimum automaton (w.r.t. the number of states) consistent with a given set of input/output pairs. Without any particular

[8] The extent of ceq in model 1 is defined by $\{(8, 12), (12, 8)\} \cup \{(x, x)\colon\ x = 1, \ldots, 12\}$.

assumptions about the environment or the agent's perceptual abilities, the problem of finding this smallest automaton is NP-complete [1,20].

5. Topological maps

Actions in the causal theory convey patterns of experience but not spatial configuration. Spatial configuration is considered by the topological theory where actions are categorized into two classes: *turns* and *travels*. Turns and travels are explained by a new ontology, that of *places* and *paths*. Turn actions leave the agent at the same place. Travel actions move the agent to a new place along a path.

Grouping places into *regions* allows an agent to reason efficiently about its spatial knowledge. Regions themselves can be grouped to form new regions forming a spatial abstraction hierarchy. (In this article we do not consider this hierarchy.) In Section 6 we define *boundary regions* associated with paths. Informally, a path has associated three disjoint regions: the set of places in the path, the set of places to the left of the path, and the set of places to the right of the path. Boundary regions allow the agent to distinguish distinctive states, for two distinctive states can be considered different if they are in different boundary regions of the same path (see Example 17).

Local metrical information derived during action execution is considered in the topological theory. For instance, the distances among places on a path or the angles among paths intersecting in a place can be accommodated in the topological map. We study the use of metrical information in Section 7.

5.1. Ontology of the topological theory

The main purpose of the topological theory $TT(E)$ is to minimize the set of topological paths and topological places consistent with the given experiences E. The concepts of *path* and *place* are used to distinguish environment states that are not distinguishable by actions and views alone. We use the predicate $\mathbf{teq}(\mathbf{ds}, \mathbf{ds}')$ to indicate that distinctive states *ds* and ds' are topologically indistinguishable. This will be the case, when in addition to not being distinguishable by views and actions, *ds* and ds' are at the same place facing the same direction along the same path.

Within the sort of places, we distinguish between *topological places* and *regions*. A topological place is a set of distinctive states linked by turn actions. A region is a set of places. We use the predicates **tplace** and **is_region** to identify these subsorts.

A path defines an order relation among places connected by travel with no turn actions. They play the role of streets in a city layout. Among paths, *topological paths* correspond to those paths whose places are topological places. We use the predicate **tpath** to identify these paths. A path connecting regions is called a **route**. A path has two directions, *pos* and *neg*, which can be thought of as referring to "upstream" and "downstream" in the order of places on the path. The path direction also serves as a frame of reference for specifying the boundary regions describing places to the left and right of the path (see Section 6). The sort of path directions is completely defined by *pos* and *neg*. For a direction *dir*, $-dir$ is defined such that $-pos = neg$ and $-neg = pos$.

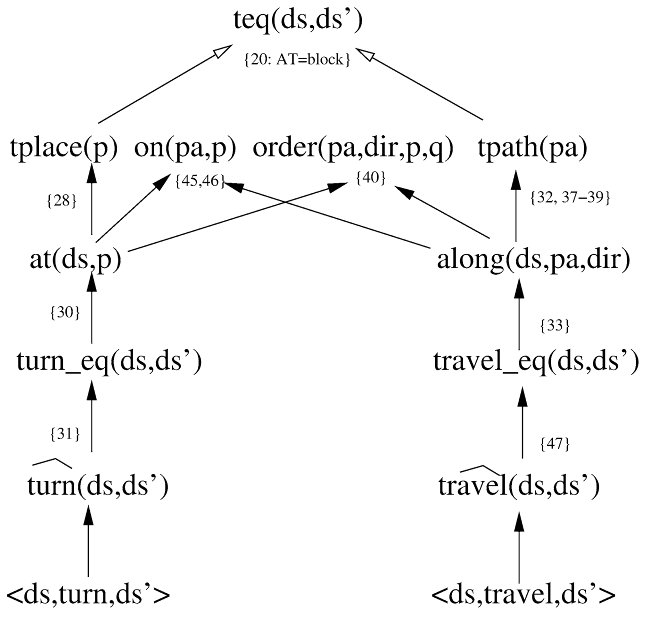

Fig. 7. Dependency among predicates in $TT(E)$. Labels on the graph's arrows refer to the axioms relating the predicates pointed by the arrows. Distinctive states related by turns modulo *teq* (*turn_eq*) must be *at* the same topological place (*tplace*). Distinctive states related by travels modulo *teq* (*travel_eq*) are *along* the same topological path (*tpath*). Knowing at which places and along which paths distinctive state are, determines what places are *on* what paths. The order of places on a path is derived from travels among distinctive states along a path. Since the extents of *travel_eq* and *turn_eq* must be defined in order to determine places and paths, one has to know what distinctive states are *teq*. The arrows pointing to *teq* on the top of the diagram indicate that among the possible interpretations for *teq*, the preferred models of the theory select those that lead to a map where a minimum set of paths and places are needed to explain the schemas at the bottom of the diagram.

The relations among distinctive states, places and paths are characterized in terms of the following predicates: **on(pa, p)**, *place p* is on *path pa*; **order(pa, dir, p, q)**, *place p* is before *place q*, when facing direction *dir* on path *pa*; **at(ds, p)**, *distinctive state ds* is at *place p*; and **along(ds, pa, dir)**, *distinctive state ds* is along *path pa* in direction *dir*. Fig. 7 summarizes the dependencies among the above predicates. Section 5.2 formalizes these relationships.

Since a map can be arbitrarily large, no finite domain can be adequate and so we require the sorts of places and paths to be contably infinite. This is not to say that the topological map has infinite number of *places* or *paths*. Given a model of the theory, the topological map corresponds to the submodel obtained by restricting the different predicates to *topological places*, *regions*, *topological paths* and *routes*. Since *topological places* are identified with finite sets of distinctive states and *topological paths* are identified with finite sequences of distinctive states, the topological map associated with a finite set of schemas (and so a finite set of distinctive states) has a finite number of *topological places* and *topological paths*. We require infinite sorts of places and paths to avoid models being non-comparable due to a mismatch in the cardinalities of the sorts, as illustrated in Example 16.

5.2. The topological theory

The topological theory associated with *E*, **TT(E)**, is the following nested abnormality theory (NATs) [36] (see Appendix A): (the condition that the sorts of places and paths are countably infinite is formalized by asserting the existence of a bijection between these sorts and the natural numbers)

$$\begin{aligned}
&TT(E) = &(19)\\
&\quad \text{there exist countably infinitely many places},\\
&\quad \text{there exist countably infinitely many paths},\\
&\quad \neg\exists p\big[tplace(p) \wedge is\_region(p)\big],\\
&\quad \neg\exists pa\big[tpath(pa) \wedge route(pa)\big],\\
&\quad COMPLETION(E),\\
&\quad \text{Axioms (2)–(10)},\\
&\quad \langle ds, a, ds'\rangle \wedge \langle ds, a, ds''\rangle \rightarrow ds' = ds'', \quad \text{(Axiom (15))}\\
&\quad T\_block,\\
&\quad AT\_block.
\end{aligned}$$

The block **T_block** defines the properties of the predicates $\widehat{turn}$, $\widehat{travel}$, and $\overrightarrow{travel}$. $\widehat{turn}$ is the equivalence closure of the schemas $\langle\cdot, turn, \cdot\rangle$; $\widehat{travel}$ and $\overrightarrow{travel}$ are the equivalence and transitive closure of the schemas $\langle\cdot, travel, \cdot\rangle$ respectively (Appendix D).

The block **AT_block** is the heart of our theory. It defines how the agent groups distinctive states into *places*, and how *places* are ordered by *paths*. The purpose of this block is to define the extent of the predicates *tpath*, *tplace*, *at*, *along*, *order*, *on* and *teq*. The block has the associated circumscription policy

$$\mathbf{circ}\ tpath \succ tplace\ \mathbf{var}\ \overrightarrow{SSHpred}$$

stating that a minimum set of topological paths is preferred to a minimum set of topological places. The symbol $\succ$ indicates prioritized circumscription (see Appendix A). $\overrightarrow{SSHpred}$ stands for the tuple of predicates $\langle \mathbf{at}, \mathbf{along}, \mathbf{order}, \mathbf{on}, \mathbf{teq}, turn\_eq, travel\_eq\rangle$. The predicates *travel_eq* and *turn_eq* are "auxiliary" predicates used in our topological theory. Although they are completely defined in terms of *teq*, $\widehat{turn}$ and $\widehat{travel}$, they need to vary in the circumscription policy. The block *AT_block* is defined as follows:

$$\begin{aligned}
&AT\_block = &(20)\\
&\{\, max\ teq:\\
&\quad teq(ds, ds),\\
&\quad teq(ds_1, ds_2) \rightarrow teq(ds_2, ds_1),\\
&\quad teq(ds_1, ds_2) \wedge teq(ds_2, ds_3) \rightarrow teq(ds_1, ds_3), &(21)\\
&\quad teq(ds_1, ds_2) \rightarrow View(ds_1, v) \equiv View(ds_2, v), &(22)\\
&\quad teq(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \rightarrow teq(ds_1', ds_2'), &(23)
\end{aligned}$$

$$teq(ds_1, ds_2) \rightarrow \forall p\big[at(ds_1, p) \equiv at(ds_2, p)\big] \wedge \\ \forall pa, dir\big[along(ds_1, pa, dir) \equiv along(ds_2, pa, dir)\big], \tag{24}$$

$$\langle ds, turn, ds' \rangle \rightarrow \neg teq(ds, ds'), \tag{25}$$

$$\langle ds, turnAround, ds' \rangle \wedge \langle ds, turnAround, ds'' \rangle \rightarrow teq(ds', ds''), \tag{26}$$

$$\langle ds_1, turnAround, ds_2 \rangle \wedge \langle ds_2, turnAround, ds_3 \rangle \rightarrow teq(ds_1, ds_3), \tag{27}$$

$$at(ds, p) \rightarrow tplace(p), \tag{28}$$

$$\exists! p\, at(ds, p), \tag{29}$$

$$turn\_eq(ds_1, ds_2) \equiv \forall p\big[at(ds_1, p) \equiv at(ds_2, p)\big], \tag{30}$$

$$\{min\ turn\_eq: \tag{31}$$

$$teq(ds_1, ds_2) \wedge teq(ds_3, ds_4) \wedge \widehat{turn}(ds_2, ds_3) \rightarrow turn\_eq(ds_1, ds_4),$$

$$turn\_eq(ds_1, ds_2) \wedge turn\_eq(ds_2, ds_3) \rightarrow turn\_eq(ds_1, ds_3)$$

$$\}$$

$$along(ds, pa, dir) \rightarrow tpath(pa), \tag{32}$$

$$\{min\, along: \tag{33}$$

$$\langle ds, travel, ds' \rangle \rightarrow \exists pa,\ dir\, \big[along(ds, pa, dir) \wedge along(ds', pa, dir)\big], \tag{34}$$

$$\langle ds, turnAround, ds' \rangle \rightarrow along(ds, pa, dir) \equiv along(ds', pa, -dir), \tag{35}$$

$$teq(ds_1, ds_2) \rightarrow along(ds_1, pa, dir) \equiv along(ds_2, pa, dir) \tag{36}$$

$$\}$$

$$along(ds, pa, dir) \wedge along(ds, pa1, dir1) \rightarrow pa = pa1 \wedge dir = dir1, \tag{37}$$

$$at(ds_1, p) \wedge at(ds_2, p) \wedge along(ds_1, pa, dir) \wedge \\ along(ds_2, pa, dir) \rightarrow teq(ds_1, ds_2), \tag{38}$$

$$\big[\langle ds, turn\_desc, ds' \rangle\ \wedge\ turn\_desc \neq turnAround\ \wedge \\ along(ds, pa, dir) \wedge along(ds', pa1, dir1)\big] \rightarrow pa \neq pa1, \tag{39}$$

$$\{min\, order: \tag{40}$$

$$\big[\langle ds, travel, ds' \rangle \wedge at(ds, p) \wedge at(ds', q) \wedge \\ along(ds, pa, dir) \wedge along(ds', pa, dir)\big] \rightarrow order(pa, dir, p, q), \tag{41}$$

$$order(pa, pos, p, q) \equiv order(pa, neg, q, p), \tag{42}$$

$$order(pa, dir, p, q) \wedge order(pa, dir, q, r) \rightarrow order(pa, dir, p, r) \tag{43}$$

$$\}$$

$$\neg order(pa, dir, p, p), \tag{44}$$

$$\{min\, on:\ \ at(ds, p) \wedge along(ds, pa, dir) \rightarrow on(pa, p)\,\} \tag{45}$$

$$on(pa, p) \wedge on(pa, q) \wedge tpath(pa) \rightarrow \tag{46}$$
$$\exists ds_1, dir_1, ds_2, dir_2 \big[at(ds_1, p) \wedge along(ds_1, pa, dir_1) \wedge at(ds_2, q) \wedge along(ds_2, pa, dir_2) \wedge travel\_eq(ds_1, ds_2)\big],$$

$$\{min\ travel\_eq: \tag{47}$$
$$\widehat{travel}(ds_1, ds_2) \rightarrow travel\_eq(ds_1, ds_2),$$
$$\langle ds_1, turnAround, ds_2\rangle \rightarrow travel\_eq(ds_1, ds_2) \wedge travel\_eq(ds_2, ds_1)$$
$$teq(ds_1, ds_2) \wedge teq(ds_3, ds_4) \wedge travel\_eq(ds_2, ds_3) \rightarrow travel\_eq(ds_1, ds_4),$$
$$travel\_eq(ds_1, ds_2) \wedge travel\_eq(ds_2, ds_3) \rightarrow travel\_eq(ds_1, ds_3)$$
$$\}$$

$$\textbf{circ}\ tpath \succ tplace\ \textbf{var}\ \overrightarrow{SSHpred} \tag{48}$$
$$\}$$

We discuss these axioms in turn.

Predicate *teq* is an equivalence relation. It stands for *topologically equal*. Whenever $teq(ds_1, ds_2)$ is the case, we can consider ds_1 and ds_2 as denoting the same environment state: ds_1 and ds_2 cannot be distinguished by views and actions (Axioms (22) and (23)), they are at the same place, and they are along the same paths (Axiom (24)).

Axiom (25) states that a *turn* action takes the agent from one distinctive state to a different one. In particular we assume that a schema of the form $\langle ds, Turn, ds\rangle$ is not included in the agent's experiences. Axiom (26) states that there is a unique (modulo *teq*) distinctive state resulting from performing a turn around action. After two turn around actions the agent is back to the same dstate (Axiom (27)). Turn around actions are special since they link distinctive states along the same path but in opposite directions (Axiom (35)).

Axioms (29) and (30) state how the agent groups distinctive states into places. Every distinctive state is at a unique topological place (Axiom (29)). Whenever the agent *turns*, it stays at the same topological place (Axiom (30)). Distinctive states grouped into a topological place should be *turn* connected (modulo *teq*) (Axiom (30)). Block (31) states that the predicate *turn_eq* corresponds to the relation $\widehat{turn}$ modulo *teq*.

Travel actions among distinctive states are abstracted to topological paths connecting the places associated with such distinctive states. Travel axioms are explained in terms of the two related predicates, *along* and *order*. Both of these predicates are the minimum ones explaining travel actions and satisfying other properties included in Blocks (33) and (40), respectively.

Block (33) defines the predicate *along*. Whenever an agent *turns around*, it stays in the same path but facing the opposite path's direction (Axiom (35)). Axiom (36) is a trivial consequence of the definition of *teq* but it has to be included in the block so that the interpretation of *along* has tuples other than the ones explicitly derived from schemas (see Example 7).

There are further restrictions on the properties of *along*. For instance, a distinctive state is along at most one path (Axiom (37)). Since Axiom (37) provides "negative" information about *along*, it does not need to be included in Block (33) (see [35, Proposition 4]). Axiom (37) prevents the existence of different paths that converge to the same distinctive state (in Section 5.3 we will make this axiom a default). Finally, Axiom (38) states that there exist at most one distinctive state indicating a path's direction at a given place on the path.

Turn actions other than *turnAround* change the path the initial and final distinctive states linked by the action are along (Axiom (39)). This axiom allows the agent to conclude the existence of different paths once it turns right or left at a place (see Example 9). This axiom prevents the existence of self-intersecting paths (Fig. 15).

Block (40) defines the predicate *order*. In addition to explaining travel actions, *order* defines an order among the places on a path satisfying the following two properties: (i) the order of places in a given path direction is the inverse of the order of places in the other path direction (Axiom (42)), and (ii), the order of places in a path is transitive (Axiom (43)).

There are further restrictions on the properties of *order*: (i) the order of places in a path should be non-reflexive (Axiom (44)), and (ii) the agent has to have traveled among the places on the same path (Axiom (46)). Since these requirements provide "negative" information about *order*, they do not need to be included in Block (40) (see [35, Proposition 4]). Notice that we rule out the existence of circular paths (Axiom (44)). In Section 5.3 we will make this axiom a default.

Axiom (46) requires the agent to have traveled among the places on the same path. *travel_eq* defines when two distinctive states are linked by travel actions without turns (except for *turnAround* actions) (see Block (47)). Example 8 illustrates how by using *travel_eq* the agent can minimize the set of topological paths.

Remark. We will be using the following properties of our theory. Axiom (37) in combination with Axioms (34), (41), and (44), imply that whenever the agent has directly traveled between two distinctive states, the places associated with these distinctive states are different.

Corollary 1. $\overrightarrow{travel}(ds, ds') \rightarrow place(ds) \neq place(ds')$, *where* $place(ds)$ *denotes the unique topological place that distinctive state ds is at* (*Axiom* (29)). *Moreover, consecutive travels among distinctive states occur along the same topological path.*

Corollary 2.

$$\overrightarrow{travel}(ds, ds') \rightarrow \exists! pa, dir\left[order(pa, dir, place(ds), place(ds')) \wedge along(ds, pa, dir) \wedge along(ds, pa, dir)\right].$$

In order to prove that distinctive states ds_1 and ds_2 are at different topological places, one has to prove that $\neg turn\_eq(ds_1, ds_2)$. The following theorem states a strong condition for when this is the case. Given an equivalence relation R, $[x]_R$ denotes the equivalence class of x according to R.

Theorem 3. *Let* ds_1 *be a distinctive state symbol such that*

$$\forall ds_2 \notin [ds_1]_{\widehat{turn}},\ [ds_2]_{teq} \cap [ds_1]_{\widehat{turn}} = \emptyset.$$

Then, $\forall ds_2 \notin [ds_1]_{\widehat{turn}},\ place(ds_2) \neq place(ds_1)$.

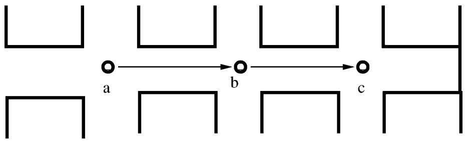

Fig. 8. Distinctive states a, b and c cannot be distinguished at the causal level (see Example 3). Using the concepts of *paths* and *places* these dstates are distinguished.

Proof. See Appendix C. □

Recall that the interpretations for *tpath* and *tplace* are finite. Our circumscription policy 48 and the fact that the sorts of paths and places are infinite implies the following fact:

Theorem 4. *Any two models of the SSH topological theory have the same number of topological paths and the same number of topological places.*

Proof. See Appendix C. □

However, Theorem 4 does not mean that a unique map is necessarily associated with a set of schemas. As shown in Example 13 the SSH topological theory could have more than one non-isomorphic model.

The next examples illustrate the interplay among the axioms in *AT_block*.

Example 5. Consider the set of experiences E

$$\langle (a, v+), \mathit{travel}, (b, v+)\rangle, \quad \langle (b, v+), \mathit{travel}, (c, v+)\rangle$$

as in Example 3, Fig. 8. From Corollary 1 we deduce that $\mathit{place}(a)$, $\mathit{place}(b)$ and $\mathit{place}(c)$ are all different places. From Corollary 2, the topological map associated with E has one topological path and three topological places. Distinctive states a and b can be distinguished though they are "causally indistinguishable" (i.e., $\mathit{ceq}(a, b) \wedge \neg \mathit{teq}(a, b)$).

Only distinctive states linked by turn actions can be grouped into a topological place (Axiom (30)). Under incomplete information this constraint could imply the existence of more places than the ones needed in a map.

Example 6. Consider the set of experiences E indicated by the formulae

$$\langle a, \mathit{travel}, b\rangle, \quad \langle b, \mathit{turnAround}, c\rangle, \quad \langle c, \mathit{travel}, d\rangle,$$

in addition to the views associated with the distinctive states. Moreover, assume that views uniquely distinguish the different distinctive states. The model for $TT(E)$ is presented in Fig. 9(c). The model has three places and one path. Not having a *turn* action relating a and d prevents the agent from grouping these distinctive states into the same place, as suggested in Fig. 9(b). Next we show why this is the case.

Since views uniquely distinguish distinctive states, then $\mathit{teq}(x, y) \equiv x = y$. From the definition of *turn_eq* (Block (31)), it follows then that $\mathit{turn\_eq} = \widehat{\mathit{turn}}$. Since the only *turn* action mentioned in E is the one in schema $\langle b, \mathit{turnAround}, c\rangle$, we deduce

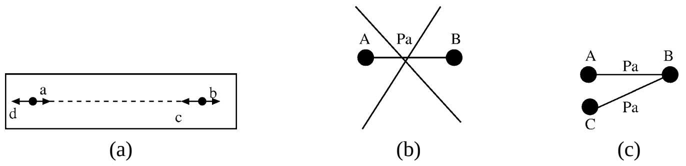

Fig. 9. (a) The agent navigates a rectangle environment getting the experiences $\langle a, travel, b\rangle$, $\langle b, turnAround, c\rangle$, and $\langle c, travel, d\rangle$. The corresponding topological map has three places and one path (c) rather than two places and one path (b). Distinctive states a and d cannot be grouped into the same topological place since they are not linked by turn actions. Notice that the order of places in the path is not total. Should the agent turn around and experience the schema $\langle d, turnAround, a\rangle$, it will consider (b) as the topological map and disregard (c).

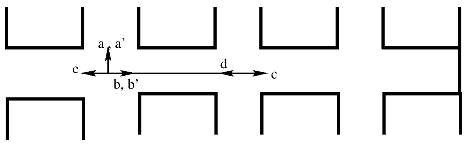

Fig. 10. The agent moves back and forth from one intersection to the other. The second time the agent visits distinctive states a and b, it gives the names a' and b'. From the topological theory it follows that these names correspond to the previously visited a and b.

that $\widehat{turn}(ds, ds') \equiv [ds = ds' \vee \{ds = b \wedge ds' = c\} \vee \{ds = c \wedge ds' = b\}]$. In particular, $\neg turn\_eq(a, d)$. In virtue of Axiom (30) we cannot conclude that a and d are at the same topological place.

The next example shows the interplay between *teq* and *along* as well as the effect of maximizing *teq*.

Example 7. Consider the set of schemas $\langle a, turnRight, b\rangle$, $\langle b, travel, c\rangle$, $\langle c, turnAround, d\rangle$, $\langle d, travel, e\rangle$, $\langle e, turnRight, a'\rangle$, $\langle a', turnRight, b'\rangle$ consistent with an agent going from one four-way intersection to another (Fig. 10). Let us consider the models of these schemas. From our axioms, at least one path and three places must exist:

Places	*Paths*
$P = \{a, b\}$	Pa: $b-c\ \ d-e$
$Q = \{c, d\}$	
$R = \{e, a', b'\}$	

Along	*teq*
$along(b, Pa, dir), along(c, Pa, dir)$	$\neg teq(a, b), \neg teq(c, d)$
$along(d, Pa, -dir), along(e, Pa, -dir)$	$\neg teq(e, a'), \neg teq(a', b')$

We know that $P \neq Q$ and $Q \neq R$. By having $teq(a, a')$, we can complete the model such that $P = R$. The maximization of *teq* will force the model to have $teq(b, b')$. By including Axiom (36) in the Block (33) we are allowed to have a model in which $teq(b, b')$ is the case. Notice that a travel action has not been performed at b' and so the schemas do not support a tuple of the form $along(b', \bullet, \bullet)$.

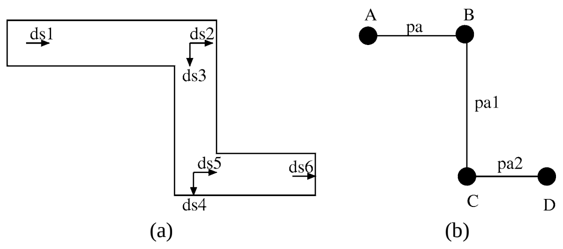

Fig. 11. By requiring the agent to have traveled among the places on a same path (Axiom (46)), different paths can be identified. (a) The agent visits the different distinctive states in the order $ds1, ds2, \ldots, ds6$; (b) depicts the topological map associated with (a). Three paths instead of only two are required to explain the agent experiences (see text).

Example 8. Consider the extension of the previous example where the schema $\langle b', \mathit{travel}, c'\rangle$ is obtained. Axiom (46) requires the agent to have traveled among places on the same path. As for places, we check this requirement "modulo" *teq*, since *teq* plays the role of equality in our theory. In this example, the agent concludes that $teq(c, c')$. Notice that $\neg\widehat{travel}(b, c')$ and $travel\_eq(b, c')$ are the case.

By requiring the agent to have traveled among the places on a same path (Axiom (46)), different paths can be identified. The next example illustrates the case.

Example 9. Suppose the agent explores the environment depicted in Fig. 11(a) obtaining the following schemas:

$$\langle ds1, \mathit{travel}, ds2\rangle \langle ds2, \mathit{turnRight}, ds3\rangle \langle ds3, \mathit{travel}, ds4\rangle$$
$$\langle ds4, \mathit{turnLeft}, ds5\rangle \langle ds5, \mathit{travel}, ds6\rangle.$$

We assume that the agent associates different views with the different distinctive states in the example. Axiom (29) implies that there exist places A, B, C and D (see Fig. 11(b)) such that

$$at(ds1, A), \quad at(ds2, B), \quad at(ds3, B), \quad at(ds4, C), \quad at(ds5, C), \quad at(ds6, D).$$

Moreover, Corollary 1 implies that $A \neq B,\ B \neq C,\ C \neq D$. Under our assumption that all distinctive states in the example have different views, it follows that $teq(ds_1, ds_2) \equiv ds_1 = ds_2$ and thus $\widehat{turn} = turn\_eq$. Since $\neg\widehat{turn}(ds1, ds3)$, $\neg\widehat{turn}(ds1, d5)$ and $\neg\widehat{turn}(ds2, ds6)$ are the case, A, B, C and D are all different. Axiom (34) implies that there exist paths Pa, $Pa1$, $Pa2$, and directions dir, $dir1$, $dir2$, such that:

$$\begin{array}{lll} order(Pa, dir, A, B), & along(ds1, Pa, dir), & along(ds2, Pa, dir), \\ order(Pa1, dir1, B, C), & along(ds3, Pa1, dir1), & along(ds4, Pa1, dir1), \\ order(Pa2, dir2, C, D), & along(ds5, Pa2, dir2), & along(ds6, Pa2, dir2). \end{array}$$

Schemas $\langle ds2, \mathit{turnRight}, ds3\rangle$ and $\langle ds4, \mathit{turnLeft}, ds5\rangle$, and Axiom (39) implies that $Pa \neq Pa1$, $Pa1 \neq Pa2$. Since $teq(ds_1, ds_2) \equiv ds_1 = ds_2$ and there is not *turnAround* schemas in E, then $\widehat{travel} = travel\_eq$. Consequently $\neg\widehat{travel}(ds1, ds4)$ and $\neg\widehat{travel}(ds1, ds5)$ are the case, and in virtue of Axiom (46) it follows that $Pa \neq Pa2$.

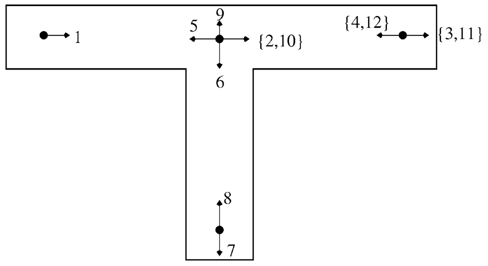

Fig. 12. The agent visits the different distinctive states in the order suggested by their numbers. The same travel action *ml* is performed when traveling from a corner to the intersection (i.e., $\langle 1, ml, 2\rangle$) and vice versa (i.e., $\langle 4, ml, 5\rangle$). A turn around action is performed when reaching a corner (i.e., $\langle 3, turnAround, 4\rangle$, $\langle 7, turnAround, 8\rangle$, etc.). Assume that the different corners have the same views (i.e., $view(1) = view(4) = view(8)$, $view(3) = view(7) = view(11)$), and views associated with the other distinctive states are different. Three different causal models can be associated with the agent exploration of this T-environment but only one of them is consistent with topological information (see text).

Example 10. Consider the same T-environment exploration presented in Example 4 (see Fig. 12). When using only causal information, three possible models are associated with the exploration. When using topological information, only one of these models is possible as illustrated next.

The three causal models associated with T-environment are:

1. Model 1: $ceq(8, 12), ceq(12, 8), ceq(x, x)$.
2. Model 2: $ceq(1, 12), ceq(12, 1), ceq(x, x)$.
3. Model 3: $ceq(4, 12), ceq(12, 4), ceq(3, 11), ceq(11, 3), ceq(2, 10), ceq(10, 2), ceq(x, x)$.

We are to show that only model 3 above is consistent with topological information. For this we show the following three facts: (i) any model must have at least 2 tpaths and 5 tplaces (since there is not a turn action between dstates $\{5, 6\}$ and dstates $\{2, 9, 10\}$, these dstates are not at the same topological place, as suggested by Fig. 12) (ii) there is a model with 2 tpaths and 5 tplaces (this is the intended model), (iii) a model of $\neg teq(2, 10)$ must have at least 6 tplaces. This last statement implies that models 1 and 2 above are not consistent with topological information.

From $\langle 1, travel, 2\rangle$ and $\langle 2, travel, 3\rangle$, Corollary 2 implies that there exist a path $Pa1$ and direction $dir1$ such that

$$along(1, Pa1, dir1), \quad along(2, Pa1, dir1), \quad along(3, Pa1, dir1).$$

Moreover, Corollary 1 implies that

$$place(1) \neq place(2), \quad place(2) \neq place(3), \quad place(1) \neq place(3).$$

From $\langle 3, turnAround, 4\rangle$, $\langle 4, travel, 5\rangle$, Axiom (35) and Corollary 2, it is the case that $along(4, Pa1, -dir1)$, $along(5, Pa1, -dir1)$. Similarly, from $\langle 5, turnLeft, 6\rangle$, $\langle 6, travel, 7\rangle$, $\langle 7, turnAround, 8\rangle$, $\langle 8, travel, 9\rangle$ we conclude that there exist a path $Pa2$ and direction $dir2$ such that $Pa1 \neq Pa2$ (Axiom (39)) and

$$place(5) \neq place(8), \quad along(6, Pa2, dir2), \quad along(7, Pa2, dir2),$$
$$along(8, Pa2, -dir2), \quad along(9, Pa2, -dir2).$$

From $\langle 9, turnRight, 10\rangle$, $\langle 10, travel, 11\rangle$, $\langle 11, turnAround, 12\rangle$, there exist path $Pa3$ and direction $dir3$ such that $Pa2 \neq Pa3$ and

$$along(10, Pa3, dir3), \quad along(11, Pa3, dir3), \quad along(12, Pa3, -dir3).$$

Theorem 3 allow us to conclude that $place(5) \notin \{place(1), place(2), place(3)\}$. The same argument shows that $place(8) \notin \{place(1), place(2), place(3), place(5)\}$. *Consequently, a minimal model of the theory must have at least two tpaths and five tplaces.*

Notice that in the intended model of the T-environment, $Pa1 = Pa3$, $dir1 = dir3$, $teq(2, 10)$, $teq(3, 11)$ and $teq(4, 12)$. This model is indeed a model of $TT(E)$ since at least two topological paths and five topological places are needed to explain E, and consequently any model must have two topological paths and five topological places (Theorem 4).

If $\neg teq(2, 10)$ were the case, then Theorem 3 allows to conclude that $place(9) \notin \{place(1), place(2), place(3), place(5), place(8)\}$ and so the model will have at least six tplaces. Consequently $teq(2, 10)$ has to be the case in a minimal model of the theory.

Example 11. Consider an extension of the previous example where we have the additional schemas $\langle 9, turnLeft, 5'\rangle$, $\langle 5', turnRight, 9\rangle$. In this case, the intended model has *four* places and two paths. Notice that now the agent can conclude that $place(5) = place(2)$ by making $teq(5', 5)$ and so $turn\_eq(5, 2)$.

The theory does not assume a "rectilinear" environment where paths intersect at most in one place. Consider the next example.

Example 12. Suppose the agent explores the environment depicted in Fig. 13 obtaining the following schemas:

$$\begin{array}{lll}
\langle ds1, turnAround, ds2\rangle & \langle ds2, turnAround, ds1\rangle & \langle ds1, travel, ds3\rangle \\
\langle ds3, turnRight, ds4\rangle & \langle ds4, turnLeft, ds3\rangle & \langle ds3, travel, ds6\rangle \\
\langle ds6, turnLeft, ds7\rangle & \langle ds7, travel, ds4\rangle & \\
\langle ds4, turnRight, ds5\rangle & \langle ds5, travel, ds2\rangle. &
\end{array}$$

We assume that views uniquely distinguish the different distinctive states. From Corollary 1 there exist the different places A, B, and C suggested in the figure. In ad-

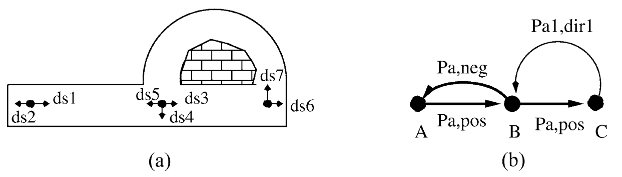

Fig. 13. The environment in (a) illustrates a case where different paths intersect at more than one place. Suppose the agent explores the environment by visiting the different distinctive states in the order $ds1, ds2, ds1, ds3, ds4, ds3, ds6, ds7, ds4, ds5, ds2$; (b) depicts the topological map associated with this environment.

dition, Corollary 2 implies the existence of a path, *Pa*, and direction, say *pos*, such that $order(Pa, pos, A, B)$, $order(Pa, pos, B, C)$, $order(Pa, pos, A, C)$. Moreover, from schemas $\{\langle ds7, travel, ds4\rangle, \langle ds5, travel, ds2\rangle\}$ and Axiom (34), there exist paths *Pa*1, *Pa*2, and directions *dir*1, *dir*2, such that

$$order(Pa1, dir1, C, B) \wedge along(ds7, Pa1, dir1) \wedge along(ds4, Pa1, dir1),$$
$$order(Pa2, dir2, B, A) \wedge along(ds5, Pa2, dir2) \wedge along(ds2, Pa2, dir2).$$

Since $along(ds6, Pa, pos)$, from Axiom (39) and schema $\langle ds6, turnLeft, ds7\rangle$ we conclude that $Pa \neq Pa1$. Since we are minimizing paths, by setting $Pa2 = Pa$ and $dir2 = neg$, we obtain a minimal model for E. Notice that in this model, places B and C belong to two different paths, *Pa* and *Pa*1.

There are some patterns of experience in which our theory is not applicable. In particular, Axiom (44) rules out circular paths and Axiom (37) rules out experiences where different paths merge into the same distinctive state. In Section 5.3 we extend the topological theory to deal with these type of paths.

Since the positive and negative direction of a path are chosen arbitrarily (Axiom (34)), there is not a unique minimal model for $TT(E)$. Given any model M of $TT(E)$ one could define another model M' of $TT(E)$ by choosing a path *pa* in M and reversing the roles of the directions *pos* and *neg* for *pa*. We will consider these "up to path direction isomorphic" models to be the same. However, no "up to path direction isomorphic" topological maps can explain the same pattern of experience. This happens because the experiences are incomplete, or the agent's sensors are weak.

Example 13. Assume that the agent visits places A, B, C, D, E, F, C in the order suggested by Fig. 14. Assume also that intersections look alike. In particular, places B and C look alike. Given this information, the agent is not able to decide whether it is back to B or C and consequently two minimal models can be associated with the set of experiences in this environment (Figs. 14(b), (c)).

Metrical information can be used to deduce the correct topology (see Example 18). However, if the agent accumulates more information, by turning at C and traveling to D, then topological information suffices to deduce that the topology of the environment is the one in Fig. 14(b). This is the case since the views at C and D are different.

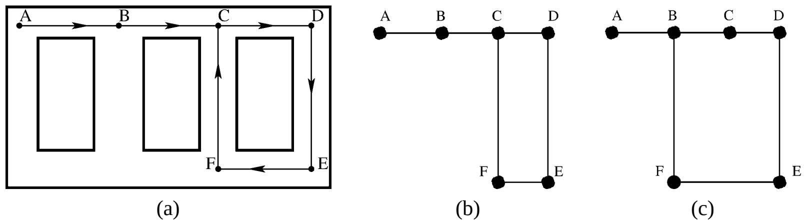

Fig. 14. (a) The agent goes around the block visiting places $A, B, \ldots, F, C$ in the order suggested in the figure. Intersections B and C look alike to the agent. (b) and (c) represent two possible representations for the environment in (a). Topological information is not enough to decide whether the agent is back to B or C.

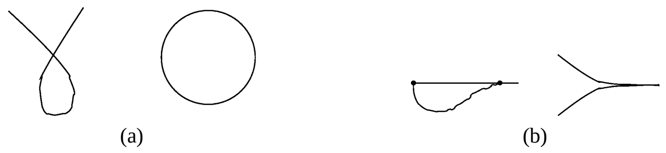
Fig. 15. (a) Self-intersecting paths. (b) Convergent paths.

5.3. Coping with self intersecting paths

The topological theory presented in the previous section is adequate for representing environments where "complex" paths configurations do not occur. In particular, we assume that self-intersecting and convergent paths do not exist (Fig. 15). In this section we extend our theory to deal with these types of paths. Converging paths are the standard counterexample for the axiom stating that distinctive states are along a unique path (Axiom (37)). We replace Axiom (37) by the block

$$\begin{aligned}
&\{\, min\, convergent\_paths: \\
&\quad \big[along(ds, pa, dir) \wedge along(ds, pa1, dir1) \\
&\quad \wedge \neg[pa = pa1 \wedge dir = dir1]\big] \rightarrow convergent\_paths(pa, pa1) \\
&\}
\end{aligned}$$

Self-intersecting paths are the standard counterexample for the axioms stating that turning changes the path (Axiom (39)), at a place there is at most one distinctive state along a path direction (Axiom (38)), and the order of places in a path is not reflexive (Axiom (44)). We replace these axioms by the block

$$\begin{aligned}
&\{min\, self\_intersecting: \\
&\quad order(pa, dir, p, p) \rightarrow self\_intersecting(pa), \\
&\quad \big[\langle ds, turn\_desc, ds'\rangle \wedge turn\_desc \neq TurnAround \wedge along(ds, pa, dir) \\
&\quad\quad \wedge along(ds', pa, dir1)\big] \rightarrow self\_intersecting(pa), \\
&\quad \big[at(ds_1, p) \wedge at(ds_2, p) \wedge along(ds_1, pa, dir) \wedge along(ds_2, pa, dir) \\
&\quad\quad \wedge \neg teq(ds_1, ds_2)\big] \rightarrow self\_intersecting(pa) \\
&\}
\end{aligned}$$

While we have defined *convergent* and *self-intersecting* paths, we still need to state that by default these kind of paths do not exist. This is accomplished by giving priority to the minimization of these two predicates over any other predicate. The new circumscription policy associated with our theory becomes

$$\mathbf{circ}\, self\_intersecting \succ convergent\_paths \succ tpath \succ tplace\, \mathbf{var}\, \overrightarrow{SSHpred}. \qquad (49)$$

The new theory is a *conservative* extension of our previous theory, since *any topological map with respect to our previous theory is a topological map according to the new theory*. In particular, the maps associated with Examples 5–12 are still valid maps for the new theory. Next we study some cases we could not handle before.

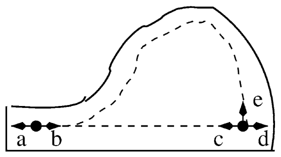

Fig. 16. Distinctive state a is along two different paths. These two paths are declared convergent paths in the model of our theory.

Example 14. Suppose the agent has experienced the following schemas (Fig. 16):

$$\langle b, \mathit{travel}, d\rangle \quad \langle d, \mathit{turnAround}, c\rangle \quad \langle c, \mathit{turnRight}, e\rangle$$
$$\langle e, \mathit{travel}, a\rangle \quad \langle a, \mathit{turnAround}, b\rangle.$$

From Axiom (34) we know that exist paths Pa, $Pa1$ and directions dir, $dir1$ such that $along(b, Pa, dir)$, $along(d, Pa, dir)$, $along(e, Pa1, dir1)$ and $along(a, Pa1, dir1)$ are the case. Moreover, from Axiom (35) it follows that $along(b, Pa1, -dir1)$. We have two possible models for these schemas:

- *Model* 1. In this model $Pa \neq Pa1$. Consequently, $self\_intersecting = false$ and $convergent\_paths(Pa, Pa1)$ are the case.
- *Model* 2. In this model $Pa = Pa1$. Consequently, $self\_intersecting(Pa)$ and $convergent\_paths = false$ are the case.

We prefer model 1 over model 2 according to the circumscription policy (49).

Example 15. Consider the set of experiences E

$$\langle (a, v+), \mathit{travel}, (b, v+)\rangle, \quad \langle (b, v+), \mathit{travel}, (c, v+)\rangle$$

as in Example 3. In the intended minimal model there are one path and three places. There are however other interpretations for the schemas. For example, the agent traveled from a to b along path Pa and then "changed" paths to travel from b back to a along path $Pa1$. In this "model" we have $teq(a, c)$, $Pa \neq Pa1$ and $convergent\_paths(Pa, Pa1)$. The model has two paths and two places (less places than the intended model). By prioritizing paths over places we get rid of this model. The prioritization conveys the heuristic that "paths help to determine places". In general if "concept X helps to determine concept Y" then X has higher priority than Y in our circumscription policy.

Our requirement of infinite *places* and *paths* allow us to compare any two models of the theory (see Theorem's 4 proof). This requirement also allow us to deal with unexpected models as illustrated in the next example.

Example 16. Consider the schema $\langle a, \mathit{travel}, b\rangle$ where a and b have the same view. The intended model has one topological path and two topological places. One expects that the path is not circular (self-intersecting), and so the existence of two places. However, without requiring the existence of enough places, the following model is also possible:

$places = \{A\}$, $tplace = \{A\}$	$paths = \{Pa\}$, $tpaths = \{Pa\}$
$teq(a, b)$	$self\_intersecting(Pa)$
$at(a, A), at(b, A)$	$along(a, Pa, pos), along(b, Pa, pos)$
	$order(Pa, pos, A, A)$

In this model, *self_intersecting*(*Pa*) must be the case, since the universe of places only has one place. Notice that when comparing two models according to the circumscription policy (49), the universe of *paths* and *places* in the models has to be the same. One can vary the interpretation of *tpath*, *tplace*, and so on, but **not** the universe of *paths* and *places*. The model above is ruled out by requiring the universe of *places* to have enough (infinite) places.

6. Boundary regions

Topological paths play the role of *streets* in a city layout map. Streets are often used as a reference for specifying the location of a given place: a place will be either on the given street or in one of the “two sides” —left or right—of the street.

Mathematically, the concept of left and right of a topological path is related to the topological one of the interior and exterior of a curve. While not all curves have a well defined interior and exterior (for example, consider a spiral, or a fractal curve), closed not self-intersecting curves—*Jordan curves*—do have associated interior and exterior sets: when the curve is removed, the plane is divided into two disjoint connected sets [3]. Moreover, in order to go from the interior to the exterior (or vice versa) of the curve γ, one has to cross γ. Our analogy of topological paths and mathematical curves breaks down because in general the agent might be able to travel from one side of the path to the other without crossing the path. This can happen because of the agent’s inability to detect that it has crossed the path, or (more often) because paths are not long enough to divide the environment into two regions (for example, consider a dead-end street).

In order to determine boundary relations—the location of a place with respect to a path—we formally state the following heuristic. Suppose the agent is at an intersection on a given path, and it then turns right. If the agent now travels, any place it finds while traveling with no turns will be on the right of the starting path. While this heuristic draws the correct conclusion in a rectilinear environment, it may draw incorrect conclusions when paths are not straight. Consequently, we state our heuristic as a “defeasible” rule so as not to conclude a boundary relation when inconsistent sources of information exist (Fig. 17).

TurnRight and *turnLeft* actions are used to define the relative orientation between paths at a given place (Section 6.1), relations that are then used to infer whether a place is on the left or the right of a given path (Section 6.2). The boundary relations inferred by an agent may not be complete: the agent does not necessarily know the location of each place with respect to each path. Nevertheless, the boundary relations inferred by the agent are useful to distinguish places otherwise not distinguishable by the topological maps as described so far (see Example 17).

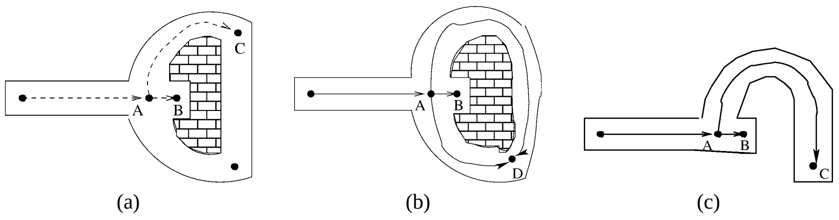

Fig. 17. Different environments illustrating how our default to determine boundary relations work. In (a) we conclude by default that place C is to the left of the path from A to B. In (b) we conclude nothing about the location of place D with respect to this path. In (c) we conclude that place C is to the left of the path from A to B. This is the case since there is no information to conclude otherwise.

6.1. Qualitative orientation of paths at a place

We extend the topological level in order to represent the relative orientation among paths that intersect at a given place. We use the predicates

$$\mathbf{totheLeftOf}(\mathbf{p}, \mathbf{pa}, \mathbf{dir}, \mathbf{pa1}, \mathbf{dir1}), \quad \mathbf{totheRightOf}(\mathbf{p}, \mathbf{pa}, \mathbf{dir}, \mathbf{pa1}, \mathbf{dir1})$$

to represent the facts that (i) *p* is a *place* on both paths, *pa* and *pa*1, and (ii), when the agent is at *place p* facing on the direction *dir* of *pa*, after executing a turn left (right) action, the agent will be facing on the direction *dir*1 of *pa*1 (see Fig. 18).

The predicates *totheLeftOf* and *totheRightOf* are derived from the actions performed by the agent at a place:

$$\begin{aligned}
&\{min\, totheRightOf,\ min\, totheLeftOf: \\
&\quad \big[\langle ds, turnRight, ds1\rangle \wedge at(ds, p) \wedge along(ds, pa, dir) \wedge along(ds1, pa1, dir1)\big] \\
&\qquad \rightarrow totheRightOf(p, pa, dir, pa1, dir1), \\
&\quad \big[\langle ds, turnLeft, ds1\rangle \wedge at(ds, p) \wedge along(ds, pa, dir) \wedge along(ds1, pa1, dir1)\big] \\
&\qquad \rightarrow totheLeftOf(p, pa, dir, pa1, dir1). \\
&\}
\end{aligned} \tag{50}$$

6.2. Left and Right of a path

A path has associated two regions: the places to the left of the path and the places to the right of the path. We use the predicates **leftOf**(**pa**, **dir**, **lr**) and **rightOf**(**pa**, **dir**, **rr**) to denote that *region lr* (*rr*) is the left (right) region of path *pa* with respect to the path's direction *dir*. The properties of these predicate are as follows:

$$\exists! lr\{leftOf(pa, dir, lr)\},\ \exists! rr\{rightOf(pa, dir, rr)\}, \tag{51}$$

$$leftOf(pa, dir, r) \equiv rightOf(pa, -dir, r), \tag{52}$$

$$\{min\, is\_region\colon\ LeftOf(pa, dir, lr) \rightarrow is\_region(lr)\}, \tag{53}$$

$$leftOf(pa, dir, lr) \wedge leftOf(pa1, dir1, lr) \rightarrow pa = pa1. \tag{54}$$

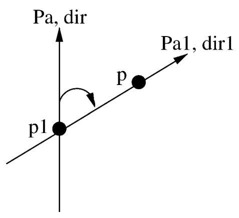

Fig. 18. Path *Pa*1 is to the right of path *Pa* at place *p*1. Place *p* is after place *p*1 on path *pa*1. By default, we conclude that place *p* is to the right of path *pa*.

Axiom (51) states the existence and uniqueness of a path's left/right regions. The domain of *leftOf* is restricted by Block (53) and Axiom (54). Since left/right regions of a path interchange when changing the path direction (Axiom (52)), constraining the domain of *leftOf* imposes similar constraints on the domain of *rightOf*.

We use the predicate **in_region(p,r)** to indicate that *place p* is in *region r*. The domain of *in_region* is constrained by Axiom (55). The properties of *in_region* are defined in Block (56). A path has associated three disjoint set of places: the places on the path, and the places to the left/right of the path (Axioms (58) and (59)). Boundary relations are derived according to Axioms (60) and (61) (see Fig. 18): (the symmetry between *leftOf* and *rightOf* defined by Axiom (52) let us write our axioms in terms of only one of these predicates)

$$\mathit{in\_region}(p, r) \to \mathit{is\_region}(r), \tag{55}$$

$$\{\, \mathit{min}\ \mathit{in\_region}: \tag{56}$$

$$\quad \{\, \mathit{in\_region}: \tag{57}$$

$$\quad \mathit{on}(pa, p) \wedge \mathit{leftOf}(pa, dir, lr) \to \neg \mathit{in\_region}(p, lr), \tag{58}$$

$$\quad \big[\mathit{leftOf}(pa, dir, lr) \wedge \mathit{rightOf}(pa, dir, rr) \wedge \mathit{in\_region}(p, lr)\big] \to \neg \mathit{in\_region}(p, rr), \tag{59}$$

$$\quad \big[\mathit{totheRightOf}(p1, pa, dir, pa1, dir1) \wedge \mathit{order}(pa1, dir1, p1, p) \wedge \mathit{rightOf}(pa, dir, rr) \wedge \neg \mathbf{Ab}(\mathbf{pa}, \mathbf{p})\big] \to \mathit{in\_region}(p, rr), \tag{60}$$

$$\quad \big[\mathit{totheLeftOf}(p1, pa, dir, dir1, pa1) \wedge \mathit{order}(pa1, dir1, p1, p) \wedge \mathit{leftOf}(pa, dir, lr) \wedge \neg \mathbf{Ab}(\mathbf{pa}, \mathbf{p})\big] \to \mathit{in\_region}(p, lr) \tag{61}$$

$$\quad \}$$

$$\}$$

Block (56) defines the extent of the predicate *in_region*. The outer preference minimizes *in_region*, so its positive instances only reflect actual observations. Normally boundary relations are false. This is the case since by default the agent does not know the location of a place with respect to a given path. The inner block (57) states under what conditions the agent can derive a boundary relation. For instance, according to Axiom (60), if at place *p*1 path *pa*1 is to the right of path *pa*, and place *p* is after *p*1 on path *pa*1, then normally it is the case that *p* is on the right of *pa* (see Fig. 18). The predicate **Ab** inside Block (57) is the *auxiliary* "abnormality" predicate associated with a NAT block (Appendix A).

(See [36] for a similar formalization of the standard example: objects normally do not fly; birds normally do.) Some sufficient conditions for when Ab is the case can be derived from Block (57) as follows.

Let *leftOf′* and *rightOf′* denote the following abbreviations:

$$\begin{aligned}&\mathit{leftOf}'(p, pa, dir) \equiv \exists lr\bigl\{\mathit{leftOf}(pa, dir, lr) \wedge \mathit{in\_region}(p, lr)\bigr\},\\&\mathit{rightOf}'(p, pa, dir) \equiv \exists rr\bigl\{\mathit{rightOf}(pa, dir, rr) \wedge \mathit{in\_region}(p, rr)\bigr\},\end{aligned}$$

which allow us to implicitly refer to the left and right regions associated with a path (these abbreviations make sense given Axiom (51)). Axioms inside Block (57) can be rewritten as follows:

$$\begin{aligned}&on(pa, p) \to \neg\,\mathit{leftOf}'(p, pa, dir) \wedge \neg\,\mathit{rightOf}'(p, pa, dir),\\&\mathit{leftOf}'(p, pa, dir) \to \neg\,\mathit{rightOf}'(p, pa, dir),\\&\bigl[\mathit{totheRightOf}(p1, pa, dir, pa1, dir1) \wedge order(pa1, dir1, p1, p) \wedge\\&\ \neg \mathit{rightOf}'(p, pa, dir)\bigr] \to Ab(pa, p),\\&\bigl[\mathit{totheLeftOf}(p1, pa, dir, dir1, pa1) \wedge order(pa1, dir1, p1, p) \wedge\\&\ \neg\,\mathit{leftOf}'(p, pa, dir)\bigr] \to Ab(pa, p).\end{aligned}$$

Using this rewriting of Block (57), one can derive the following (among others) sufficient conditions to deduce *Ab*:

$$\begin{aligned}&on(pa, p) \wedge \bigl[\mathit{totheRightOf}(p1, pa, dir, pa1, dir1) \wedge\\&\qquad\qquad order(pa1, dir1, p1, p)\bigr] \to Ab(pa, p),\end{aligned}\tag{62}$$

$$\begin{aligned}&\bigl[\mathit{totheRightOf}(p1, pa, dir, pa1, dir1) \wedge order(pa1, dir1, p1, p) \wedge\\&\ \mathit{totheLeftOf}(p2, pa, dir, pa2, dir2) \wedge order(pa2, dir2, p2, p)\bigr]\\&\qquad\qquad \to Ab(pa, p),\end{aligned}\tag{63}$$

$$\begin{aligned}&\bigl[\mathit{totheRightOf}(p1, pa, dir, pa1, dir1) \wedge order(pa1, dir1, p1, p) \wedge\\&\ \mathit{totheRightOf}(p2, pa, -dir, pa2, dir2) \wedge order(pa2, dir2, p2, p)\bigr]\\&\qquad\qquad \to Ab(pa, p).\end{aligned}\tag{64}$$

Conditions (62)–(64) show sufficient conditions for when *Ab* is the case, and consequently when the agent should not deduce boundary relations. (Condition (64) uses the symmetry between *leftOf′* and *rightOf′* defined by Axiom (52).) These conditions are in terms of predicates others than *in_region, leftOf* and *rightOf* whose extent is the purpose of Blocks (56) and (57).

6.3. Adding boundary relations to the topological map

We update the topological theory by including Axioms (50)–(61) inside the block **AT_block** (Section 5.2), and the new circumscription policy becomes

$$\mathbf{circ}\,\neg\mathbf{in\_region} \succ \mathit{tpath} \succ \mathit{tplace}\ \mathbf{var}\ \mathit{new}\overrightarrow{\mathit{SSH}}\mathit{pred}$$

where $\textit{newS}\vec{S}\textit{Hpred}$ stands for the tuple of predicates

$$\begin{array}{l}\langle\ \ \textit{at}, \textit{along}, \textit{order}, \textit{on}, \textit{teq}, \textit{turn\_eq}, \textit{travel\_eq},\\ \quad \textbf{totheRightOf}, \textbf{totheLeftOf}, \textbf{leftOf}, \textbf{rightOf}, \textbf{is\_region}\\ \rangle.\end{array}$$

The circumscription policy states that Axioms (60) and (61) should be used to draw conclusions even at the expense of having more paths or more places on the map. This is achieved by maximizing *in_region* over *tpath* in the circumscription policy. This policy also prevents the theory from preferring pathological *tpaths* and *tplaces*. By maximizing the extent of *in_region* at the expense of having possibly more paths or more places, boundary relations determine distinctions among environment states that could not be derived from the connectivity of places alone. The next example illustrates the case.

Example 17. Consider an agent visiting the different corners of a square room in the order suggested by Fig. 19(a). In addition, suppose the agent's sensory apparatus allows it to define *views* by characterizing the direction of walls and open space. Accordingly, the agent experiences *four* different views, $v1$–$v4$, in this environment.

The agent's experiences, *E,* in this environment are:

$$\begin{array}{l}\textit{View}(ds1, v1),\ \textit{View}(ds2, v2),\ \textit{View}(ds3, v1),\ \textit{View}(ds4, v2),\ \textit{View}(ds5, v1),\\ \langle ds1, \textit{turnRight}, ds2\rangle,\ \langle ds2, \textit{travel}, ds3\rangle,\ \langle ds3, \textit{turnRight}, ds4\rangle,\ \langle ds4, \textit{travel}, ds5\rangle.\end{array}$$

Suppose that the agent does not use boundary regions when building the topological map. From $\langle ds3, \textit{turnRight}, ds4\rangle$ and Axiom (39) we can deduce that $Pa \neq Pb$ in Fig. 19(b). Then the minimal topological model associated with E has two paths and two places. In this model, $\textit{teq}(ds1, ds5)$ is the case. The environment looks perfectly symmetric to the agent (Fig. 19(b))!!

Suppose now that the agent relies on boundary regions. Let P, Q, R, be the topological places associated with $d1$, $d3$ and $d5$ respectively. From Axiom (34), let Pa, Pb, dir_a and dir_b be such that

$$\begin{array}{lll}\textit{order}(Pa, dir_a, P, Q), & \textit{along}(ds2, Pa, dir_a), & \textit{along}(ds3, Pa, dir_a),\\ \textit{order}(Pb, dir_b, Q, R), & \textit{along}(ds4, Pb, dir_b), & \textit{along}(ds5, Pb, dir_b),\end{array}$$

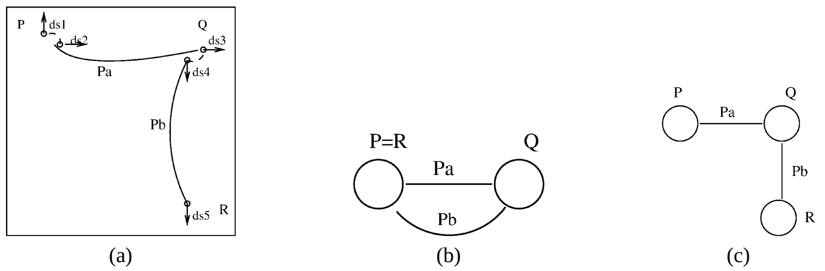

Fig. 19. (a) Sequence of actions followed by an agent while navigating a square room. Starting at distinctive state $ds1$, distinctive states are visited in the order suggested by their number. (b) and (c) depict the resulting topological map without and using boundary regions, respectively.

are the case. From Block (50) we conclude that $\mathit{totheRightOf}(Q, Pa, dir_a, Pb, dir_b)$. In the proposed model, the extent of *in_region* is maximized by declaring $Ab = \mathit{false}$ inside Block (57) and consequently (Axiom (60)) $\mathit{in\_region}(R, \mathit{right}(Pa, dir_a))$ where $\mathit{right}(Pa, dir_a)$ denotes the right region of *Pa* when facing dir_a (Axiom (51)). Moreover, from Block (56) we deduce $\mathit{in\_region}(p, r) \equiv [p = R \wedge r = \mathit{right}(Pa, dir_a)]$. Finally, from Axiom (58) we conclude $P \neq R$ since $\mathit{on}(Pa, P)$ is the case. The resulting topological map is depicted in Fig. 19(c).

Boundary relations are in general not enough to distinguish different environment states. This is the case when the agent has weak sensors, the environment is symmetric, or the agent's experiences are incomplete (see Example 19). The use of local metrical information could help on those cases although metrical uncertainty could render this extra information useless. We discuss this issues in the next section.

7. Using local metrical information

Action executions have associated metrical information representing the observed magnitude of the action. For instance, after traveling the agent may have an estimate of the distance between the "end places" of the travel action, and after turning, the agent may have an estimate of the angle turned. Different kind of metrical estimates could be associated with a travel or turn action. For example, the agent could measure the arc length associated with a travel action. In addition, it could measure the minimum distance to an object on the left and the right sides at each point along the trajectory associated with a travel action [31].

Action's executions local metrical information is integrated into frames of reference associated with topological objects:

- Each path has associated a one dimensional frame of reference which assigns a position to each place in the path.
- Each place has associated a radial frame of reference which assigns a heading (angle) to each path the place belongs to.
- Regions or places might have associated two dimensional frames of reference which assign real valued tuples to certain places. Local analog maps [4,15,60] can also be associated with places [31].

As *positions* and *headings* are derived from noisy data, there is uncertainty associated with their real values. Different representations for this uncertainty are possible: *intervals*, *probability distribution functions*, etc. As the agent repeatedly navigates among the same places and paths, new measure estimates are taken into account to update the uncertainty associated with positions and headings. In order to propagate uncertainty about the real value of positions and headings we use the compound and merge operations [56]. These operations take different forms depending on how one represents uncertainty as well as on the dimensionality of the variables' domains. In our current work we use *intervals* to represent uncertainty in position and headings, and the compound and merge operations correspond to add and intersect intervals, respectively. Nevertheless, the discussion in this section applies to other forms of representing uncertainty as long as the compound and merge operations are provided for that representation.

We use the predicate **action_execution**(s, *Int*) to state that the *interval Int* represents an estimate of the metrical information about the execution of the action associated with schema *s*. We use the notation $\langle \mathbf{ds}, (\mathbf{type\,Int}), \mathbf{ds'} \rangle$, where *type* is *travel* or *turn,* as an abbreviation for the formula

$$\exists s, a\,\big\{CS(s, ds, a, ds') \wedge action\_type(a, type) \wedge action\_execution(s, Int)\big\}.$$

How the estimates are to be interpreted depends on the type of action (turn or travel) the schema refers to. In the next sections we will describe how to do so.

7.1. One dimensional frames of reference

A path has associated a one dimensional frame of reference which assigns a location to each place on the path. This location is a real number, representing the "distance" with respect to an arbitrary but fixed place on the path. This real value represents a quantity whose magnitude is derived by the robot while navigating the environment. The units of this quantity can be *meters*, *feet*, or *number of wheel rotations*. Hereafter, we assume that all quantities are given in the same units.

The distance among places on a path are derived from estimates acquired when traveling among places on the path. These estimates have to be *consistent* so that positions can be associated with places. Next we formalize these ideas.

The position of a place on a path is represented by the predicate **position1**(**path**, **place**, **position**). Positions along a path are unique and only assigned to places belonging to the path:

$$position1(pa, p, pos) \wedge position1(pa, p, pos') \rightarrow pos = pos', \tag{65}$$

$$position1(pa, p, pos) \rightarrow on(pa, p). \tag{66}$$

The distance between two places in a path is defined as the absolute value of the difference between their corresponding positions on the path. The predicate **path_distance**(**pa**, **p**, **q**, **d**) represents the fact that the distance between places *p* and *q* on path *pa* is *d*. The predicate *path_distance* is defined as follows:

$$\begin{aligned} & path\_distance(pa, p, q, d) \equiv \qquad (67)\\ & \exists pos_p, pos_q\,\big\{position1(pa, p, pos_p) \wedge position1(pa, q, pos_q) \wedge \\ & \qquad\qquad d = |pos_p - pos_q|\big\}. \end{aligned}$$

Estimates of the distance between places on a path are gathered while the agent navigates the environment. The predicate $\mathbf{path\_distance}^{\approx}(\mathbf{pa}, \mathbf{p}, \mathbf{q}, \mathbf{I_d})$ represents the fact that the closed interval I_d is an estimate of the distance between places *p* and *q* on path *pa*. Distance estimates are derived from experiences of the robot in the environment. Distance estimates are "*compounded*" to derive new estimates from known ones. Formally,

$$\begin{aligned}
&\{\,\mathit{min}\,\mathit{path\_distance}^{\approx}: \\
&\quad \big[\langle ds, (\mathit{travel}\, I_d), ds'\rangle \wedge at(ds, p) \wedge at(ds', q) \wedge \mathit{along}(ds, pa, dir) \wedge && (68)\\
&\quad\ \mathit{along}(ds', pa, dir)\big] \rightarrow \mathit{path\_distance}^{\approx}(pa, p, q, I_d), \\
&\quad \big[\mathit{order}(pa, dir, p, q) \wedge \mathit{order}(pa, dir, q, r) \wedge \mathit{path\_distance}^{\approx}(pa, p, q, I_{pq}) \wedge && (69)\\
&\quad\ \mathit{path\_distance}^{\approx}(pa, q, r, I_{qr})\big] \rightarrow \mathit{path\_distance}^{\approx}(pa, p, r, I_{pq} + I_{qr}) \\
&\}
\end{aligned}$$

where the addition of intervals is defined in the usual way: $[a, b] + [c, d] = [a + c, b + d]$. Finally, distance estimates are "*merged*" in order to have the "best" estimate associated with a distance. The predicate $\mathit{path\_distance}^{\otimes}(pa, p, r, I_d)$ denotes the merging of distance estimates:

$$\mathit{path\_distance}^{\otimes}(pa, p, r, I) \equiv_{def} I = \bigcap \{I_{est}\colon \mathit{path\_distance}^{\approx}(pa, p, q, I_{est})\}. \quad (70)$$

The distance between places on a path must be *compatible* with all of its estimates. Formally,

$$\mathit{path\_distance}^{\otimes}(pa, p, q, I_d) \rightarrow \exists d \in I_d\, \mathit{path\_distance}(pa, p, q, d). \quad (71)$$

When the agent has distance estimates available, $\mathit{path\_distance}^{\otimes}(pa, p, q, I_d)$ is always the case for some interval I_d. In a topological map $I_d \neq \emptyset$ (Axiom (71)) and it should be possible to assign locations to places on a path as specified by Axiom (67). The actual values of positions are not that important (there could be many ways to satisfy the metrical constraints). Their main use is to rule out possible interpretations of the theory where such positions do not exist given Axiom (71).

7.2. Radial frames of reference

Each place has a local frame of reference w.r.t. which path headings are associated. This information is represented by the predicate **radial**(**p**, **pa**, **dir**, **h**) denoting the fact that *when the agent is located at place p, path pa could be followed in direction dir by facing the heading h w.r.t. the radial frame of reference local to p*. Headings take values in $[0, 2\pi)$. The formalization of radial frames of reference follows the same steps as for one dimensional frames of reference. Estimates of the angle between paths at a place are gathered from *turn* actions. Angle estimates are compounded and merged as we did for distances among places in a path. We use the predicates **angle**(**p**, **pa**, **dir**, **pa1**, **dir1**, **ang**), *ang* is the angle the agent will have to turn to face path *pa*1 in direction *dir*1 when it is at place *p* facing path *pa* in direction *dir*; $\mathbf{angle}^{\approx}(\mathbf{p}, \mathbf{pa}, \mathbf{dir}, \mathbf{pa1}, \mathbf{dir1}, \mathbf{I_{ang}})$, I_{ang} is an estimate of the angle at place *p* between path *pa* in direction *dir* and path *pa*1 in direction *dir*1.

7.3. Two dimensional frames of reference

While radial and one dimensional frames of reference are associated with any place and path, respectively, there is not a general topological theory asserting when to create a two dimensional frame of reference, what places should be included in a such frame of reference, or how to assign place locations consistent with the estimates of distances and angles gathered by the agent. Having a global frame of reference including all places in the map is usually inappropriate since the uncertainty associated with some places' locations in such a frame of reference may not allow the agent to draw useful conclusions. Instead, the agent can have multiple frames of reference as well as relations among the different frames of reference [31,40]. As the agent explores the environment, new frames of reference are created when the current's location uncertainty with respect to the current frame of reference is larger than a given threshold [16,41].

The problem of assigning locations to places given some metrical constraints can be solved by borrowing methods from different fields. For example, estimation theory tells us how to estimate the true value of a given set of variables given noisy observations of the relations between those variables [18,56]. The robotics community has developed algorithms to solve a network of spatial relations [12–14,41]. Techniques from multidimensional scaling [6] and nonlinear programming [45] can also be used.

A topological map does not explicitly represent the distance or direction between two arbitrary places. In order to do so, distances between places on a path as well as the angles between paths at a place must be combined. We use the predicate $\mathbf{location2(p, q, l)}$ to indicate that the location of place q with respect to the two dimensional frame of reference associated with place p is l (a real valued pair).

When restricted to environments with "straight" paths, it is possible to state when a two dimensional frame of reference is *compatible* with the actual experiences of the robot. The next axioms state this requirement:

$$\begin{aligned} &\mathit{location2}(p, p1, l_{p1}) \wedge \mathit{location2}(p, p2, l_{p2}) \wedge \mathit{path\_distance}^{\otimes}(pa, p1, p2, I_d) \\ &\quad \rightarrow |l_{p1} - l_{p2}| \in I_d, \end{aligned} \tag{72}$$

$$\begin{aligned} &\big[\mathit{location2}(p, p1, l_{p1}) \wedge \mathit{location2}(p, p2, l_{p2}) \wedge \mathit{location2}(p, p3, l_{p3}) \wedge \\ &\ \mathit{order}(pa, dir, p1, p2) \wedge \mathit{order}(pa', dir', p2, p3) \wedge \\ &\ \mathit{angle}^{\otimes}(p2, pa, dir, pa', dir', I_{ang})\big] \\ &\quad \rightarrow \mathit{angle}(-\overrightarrow{l_{p2}l_{p1}}, \overrightarrow{l_{p2}l_{p3}}) \in I_{ang}, \end{aligned} \tag{73}$$

where $\mathit{angle}(\vec{v}, \vec{w})$ denotes the angle in $[0, 2\pi)$ from vector $\vec{v}$ to vector $\vec{w}$. When curved paths are possible, the predicate *path_distance* represents distance *along the path*, not straight-line distance between end point. To handle curved paths, we have to separate those two concepts, or have estimates of both types of "distances".

Axioms (72) and (73) assume that paths are straight. In order to deal with more general paths, one should include some parameters describing the shape of the path, or at least an estimate of the change in heading while traveling [27,42]. For instance, in [27] travel actions were represented as $\langle ds, (\mathit{travel}\, \mathit{dist}\, \triangle\theta), ds'\rangle$, where *dist* corresponds to the distance between the places associated with ds and ds', and $\triangle\theta$ corresponds to the change of orientation while traveling. However, there is not a statement of how this extra information is used or whether it suffices to describe appropriate metrical constraints for two dimensional frames of reference. While a more detailed account of the use of metrical information is desirable, including representing and reasoning about a path's shape, we have left this description outside the scope of this work.

Using different metrical estimation approaches requires a reworking of the axioms in this section. In such case, the compound and merge operations (Axioms (68)–(70)) should be described differently. It is not difficult to define compound and merge operations for Gaussian representations of metrical uncertainty. More care will be required to update Axiom (71) which is used to refute inconsistent hypotheses, since no combination of Gaussians is logically inconsistent. A greater change will be needed in order to take into account the shape of paths when creating two dimensional frames of reference. Nevertheless, the presented axiomatization defines where in the theory the metrical information comes into place and suggests the type of axioms that need to be added.

7.4. Combining topological and metrical information

In this section we formally state what it means for the topological map to be consistent with a given set of frames of reference. In order to do so, given distinctive states $ds, ds_1, \ldots, ds_n$, we introduce the notation $\langle \mathbf{ds} : \mathbf{ds_1}, \ldots, \mathbf{ds_n} \rangle$ to state that the places associated with the different ds_i have a location in the two dimensional frame of reference associated with *ds*'s place,

Definition 3. Let $ds, ds_1, \ldots, ds_n$ be a set of distinctive states. By definition,

$$\begin{aligned} &\langle ds\colon\ ds_1, \ldots, ds_n \rangle \equiv_{def} \qquad (74) \\ &\quad \exists p \left\{ at(ds, p) \wedge \bigwedge_{i=1}^{n} \exists p_i, l_i \left[at(ds_i, p_i) \wedge location2(p, p_i, l_i) \right] \right\}. \end{aligned}$$

By **2D_Frames** we denote the formula specifying any two dimensional frames of reference used by the agent. Without loss of generality, we require two dimensional frames of reference to be specified as in definition (74). We require any model of the SSH to have only the two dimensional frames of reference specified in $2D\_Frames$. In addition, the places belonging to a frame of reference should be only those explicitly stated in (74). These last two requirements can be stated as follows:

$$\{min\, location2\colon\ 2D\_Frames\}. \qquad (75)$$

The topological theory includes local metrical information by adding Axioms (65) to (75) inside the block **AT_block** (Block (20)). The priority of predicates in the circumscription policy associated with *AT_block* remains the same. The predicates varied in the circumscription policy now include those predicates use to describe metrical information: *radial*, *position*1, *position*2, *path_distance*, $path\_distance^{\approx}$, $path\_distance^{\otimes}$, *angle*, $angle^{\approx}$ and $angle^{\otimes}$.

The next examples illustrate how metrical information is used to disambiguate the topological map.

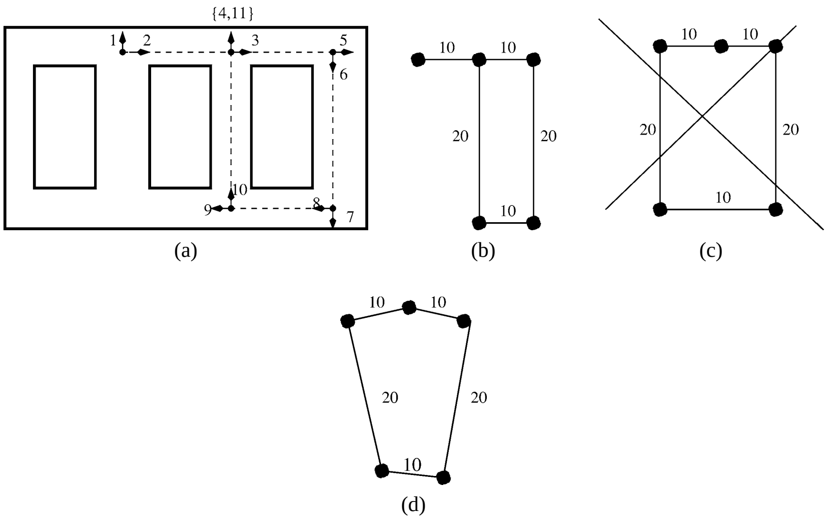

Fig. 20. (a) The robot goes around the block visiting distinctive states *ds*1 to *ds*11 in the order suggested by the figure. Distinctive state *ds*11 is observed at the same environment state as *ds*4. Assume distinctive states *ds*1 and *ds*4 look alike to the agent. (b) and (c) represent two possible topological maps for the environment in (a) (see Example 13). The model in (c) can be discarded as it is not consistent with the available metrical information. (d) With $\pm 10°$ noise associated with turn actions, the agent cannot use metrical information to discard the environment depicted in (c).

Example 18. Consider Example 13 where two topological maps are consistent with the agent's experiences (see Fig. 20). Suppose that "perfect" metrical information is available to the agent.

How does the agent figure out that it is back to *ds*4 rather than to *ds*1? As claimed in Example 13 both options $teq(ds4, ds11)$ and $teq(ds1, ds11)$ are topologically possible (Figs. 20(b), (c)). However, given the metrical information above, only the assumption $teq(ds4, ds11)$ is a consistent one. To deduce this fact, the agent includes the frame of reference $\langle ds4\colon ds1, \ldots, ds11\rangle$ in E, which renders impossible $teq(ds4, ds11)$.

Should the metrical information have been less precise, the agent might not benefit from this extra metrical information. For example, suppose that instead of sharp 90° turn angles, there exists a $\pm 10°$ uncertainty associated with the turn actions above (i.e., consider replacing $\langle ds1, (turn - 90°), ds2\rangle$ by $\langle ds1, (turn[-110°, -80°], ds2\rangle)$.[9] In this case the agent cannot use metrical information to deduce that it is back to *ds*4 and it will have two topological maps consistent with its information.

The example above may suggest that metrical information is used to check whether an already built topological map is consistent with metrical information. However, by including Axioms (65)–(75) inside *AT_block*, metrical information is used while building the topological map. As the next example illustrates, this may imply that the agent identifies more places than it does when not using metrical information.

[9] Whenever we use a number x instead of an interval, it is an abbreviation for $[x, x]$.

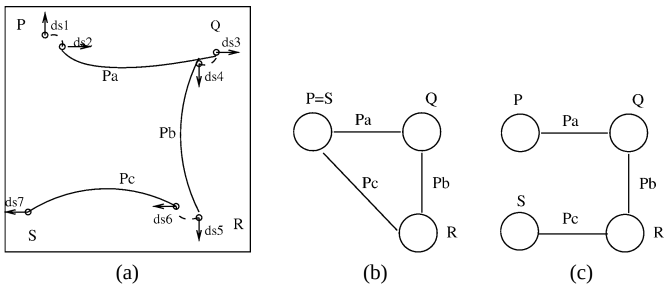

Fig. 21. (a) The agent visits distinctive states $ds1$ to $ds7$ by the order suggested in the figure. Suppose all corners look alike to the agent. In particular, $ds1$ and $ds7$ share the same view. (b) Topological map associated with (a) when metrical information is not available. (c) Topological map associated with (a) when metrical information is available. In this case, the places associated with $ds1$ and $ds7$ are different ($P \neq S$).

Example 19. Consider an agent visiting the different corners of a square room in the order suggested by Fig. 21(a). In addition, suppose the agent's sensory apparatus allows it to define *views* by characterizing the direction of walls and open space so that all corners look alike to the agent (see Example 17). Suppose the agent has access to perfect metrical information and uses it while building the metrical map.

In order to decide whether the agent is back to $ds1$, the frame $\langle ds1\colon ds1, \ldots, ds7 \rangle$ is created. Given the available metrical estimates it is not possible to have $teq(ds1, ds7)$ while satisfying the metrical constraints. Consequently, the topological map will have *four* places instead of *three*, as illustrated in Fig. 21(c).

While in the examples above all visited distinctive states were included in a two dimensional frame of reference, this is in general not the case. In the presence of metrical uncertainty, a global frame of reference may not provide useful information to determine whether two places are the same, or to estimate the distance between two arbitrary places.

8. Algorithms

In this section we present an algorithm for calculating the topological maps associated with a set of experience E. The models associated with the causal theory (Section 4) can be calculated as the answer sets [19] of a logic program. This logic program is implemented in Smodels [43] as illustrated in [48]. It is possible to calculate the topological maps by a similar logic program. However, the number of grounding rules associated with such a program turns out to be prohibitive for practical applications.

The algorithm for calculating topological maps (the models of $TT(E)$) is stated as a *"best first"* search. A search state is implemented by a partial model, *pmodel*. A partial model of $TT(E)$ is a model of $TT(E')$, for some $E' \subseteq E$ (Section 8.1). Branches in the search are represented by creating *extensions* for the current search state (pmodel). That *pmodel′* is an extension of *pmodel* implies that *pmodel′* inherits from *pmodel* all known objects and facts.

```
Find-Models(S)
{
  ;; $S = s_0, \dots, s_n$; sequence of schemas such that $result(s_i) = context(s_{i+1})$
  $queue = \emptyset$;      $models = \emptyset$;
  $pmodel = create\text{-}new\text{-}pmodel(S)$;     $insert(pmodel, queue)$;
  while $queue \neq \emptyset$ do
    begin
      $pmodel = get\text{-}next\text{-}pmodel(queue)$;
      $s = get\text{-}next\text{-}schema(pmodel)$;
      $Explain(pmodel, s)$;
      if ($inconsistent(pmodel) \vee has\text{-}extensions(pmodel)$) then skip;
      else if $total\text{-}model(pmodel)$ then $insert(pmodel, models)$;
      else $insert(pmodel, queue)$;
    end
  return $models$;
}
```

Fig. 22. Best first search algorithm used to calculate the models of $TT(E)$. The queue contains *consistent* partial models (pmodels) to be expanded. At each step of the search, a minimal partial model is picked and the next schema from its list of associated schemas is explained. A pmodel has extensions when a branch has been created while explaining a schema. A pmodel is a *total-model* when it has no more schemas to explain. Fig. 23 defines how a pmodel explains a schema and when extensions are created.

At each step of the search a schema $\langle ds, a, ds' \rangle$ has to be explained. Either the identity of ds' can be proved or a search branch is created for every previously known distinctive state ds'_i that cannot be proven to be different from ds'. The identity of the schema's context (i.e., ds in $\langle ds, a, ds' \rangle$) is known at each step in the search.

In the branch where $teq(ds'_i, ds')$ is the case, $\neg teq(ds'_j, ds')$, $i \neq j$ are also asserted. An additional branch is created where $\neg teq(ds', ds'_j)$ are asserted. This branch represents the possibility that ds' is indeed different from previously known dstates. The next state to explore is the one that is minimal according to the order associated with the circumscription policy for $TT(E)$. This search algorithm is described in Figs. 22 and 23.

The three key steps in the search are (Fig. 23): creating a set of possible candidates to branch (*possible-equal-dstates*), generating a set of extensions when needed (*create-possible-extensions*), and explaining a schema in a given partial model (*assert-schema*). Another important issue is to detect when a partial model becomes inconsistent. We use the predicate *inconsistent*(*pmodel*) to denote this fact and the rules

$$x \overset{pmodel}{=} y \wedge x \overset{pmodel}{\neq} y \rightarrow inconsistent(pmodel),$$

$$teq(x, y) \in pmodel \wedge \neg teq(x, y) \in pmodel \rightarrow inconsistent(pmodel).$$

In the next sections we will show how to rewrite the axioms in the topological theory so they can be fed to a theorem prover to deduce equality and inequality relations. We use the rule-based system Algernon [8] as our theorem prover. In Section 8.2 we present an illustrative trace of the algorithm.

```
Explain(pmodel, s)
{ ;; $s$ is a schema $\langle ds, a, ds'\rangle$
  $candidates = \{\}$;
  if $known\text{-}result(pmodel, s)$
  then $Assert\text{-}schema(pmodel, s)$;
  else begin
    $candidates = possible\text{-}equal\text{-}dstates(pmodel, s)$;
    if $candidates \neq \{\}$
       then $create\text{-}possible\text{-}extensions(pmodel, s, candidates)$
       else $Assert\text{-}schema(pmodel, s)$
    end
}
Known-result(pmodel, s)
{ ;; $s$ is a schema $\langle ds, a, ds'\rangle$
  ;; The notation $obj \in pmodel$ indicates that object $obj$ is
  ;; known in the partial model $pmodel$.
  return $ds' \in pmodel \vee \exists\, ds^*, ds'^* \in pmodel\, [\langle ds^*, a, ds'^*\rangle \in pmodel \wedge teq(ds^*, ds)]$;
}
Assert-schema(pmodel, s)
{ ;; $s$ is a schema $\langle ds, a, ds'\rangle$. $ds$ is known in $pmodel$
  assert $s \in pmodel$;
  if $\neg$ $known\text{-}result(pmodel, s)$
    then begin
       assert $ds' \in pmodel$;
       Create places and paths needed to explain $s$.
    end
    else begin
    pick $ds'^*$ s.t. $\exists ds^* \in pmodel\, \big[teq(ds^*, ds) \wedge \langle ds^*, a, ds'^*\rangle \in pmodel\big]$;
    assert $ds' \overset{pmodel}{=} ds'^*$ in $pmodel$;
    end
}
```

Fig. 23. Explaining a schema. $known\text{-}result(pmodel,\ s = \langle ds, a, ds'\rangle)$ is the case when the equality class for ds' can be deduced in the partial model $pmodel$. $Possible\text{-}equal\text{-}dstates(cntx,\ s)$ returns dstates known in $pmodel$, having the same view as ds' and that cannot be proven different from ds' in $pmodel$. For each $ds'' \in candidates$, $create\text{-}possible\text{-}extensions(pmodel,\ s,\ candidates)$ creates an extension of $pmodel$ where $teq(ds', ds'')$ is the case. If the identity of ds' can be established, then s is asserted in $pmodel$. This declares ds' to be known in $pmodel$ and creates the places and paths that explain s according to the axioms of the topological theory $TT(E)$.

8.1. Implementation

Our logic for partial models takes the basic ideas developed in the area of formal reasoning about contexts [39]. In addition to a list of schemas to explain, a partial model has associated a set of objects (i.e., distinctive states, schemas, places, paths) that are known in the model. The basic relation among pmodels is the one of *extensions*. That $pmodel'$ is an extension of *pmodel* implies that all known objects and facts in *pmodel* are known objects and facts in $pmodel'$ (i.e., $pmodel'$ inherits from *pmodel* all known objects and facts). This inheritance property of extensions can be implemented in Algernon by rules like the next one:

$$at(ds, place, pmodel) \wedge extension(pmodel, pmodel1) \rightarrow at(ds, place, pmodel1).$$

Create candidates. *Possible-equal-dstates(pmodel,$s = \langle ds, a, ds' \rangle$)* returns a list of states that are possible equal to ds'. These are dstates known in *pmodel*, having the same view as ds' and that cannot be proven different from ds' in *pmodel*. Given s, we filter out ds'' as equal to ds' using rules including:

$$s = \langle ds, turn, ds' \rangle \wedge at(ds, p) \wedge at(ds'', q) \wedge p \neq q \rightarrow \neg teq(ds', ds'') \tag{76}$$

$$\big[s = \langle ds, travel, ds' \rangle \wedge along(ds, pa, dir) \wedge along(ds'', pa1, dir1) \wedge \neg\big[pa = pa1 \wedge dir = dir1\big]\big] \rightarrow \neg teq(ds', ds'')$$

$$\big[s = \langle ds, travel, ds' \rangle \wedge along(ds, pa, dir) \wedge at(ds, p) \wedge at(ds'', q) \wedge order(pa, dir, q, p)\big] \rightarrow \neg teq(ds', ds'').$$

The rules above are derived from the axioms in our theory. For instance, rule (76) is derived from the fact that each distinctive state is at a unique place, and distinctive states that are related by turn actions are at the same place. In the implementation, all the topological predicates have a last extra argument for a pmodel. For instance, instead of writing $at(ds, p)$ we write $at(ds, p, pmodel)$. $at(ds, p, pmodel)$ is the case when $at(ds, p)$ is true in the partial model *pmodel* (i.e., $pmodel \models at(ds, p)$).

Equality relations among topological objects (i.e., dstates, places, paths) are proved using rules derived by rewriting topological axioms. These rules include:

$$view(ds_1, v_1) \wedge view(ds_2, v_2) \wedge v_1 \neq v_2 \rightarrow \neg teq(ds_1, ds_2) \tag{77}$$

$$\langle ds, turn, ds' \rangle \rightarrow \neg teq(ds, ds'), \tag{78}$$

$$order(pa, dir, p, q) \rightarrow p \neq q, \tag{79}$$

$$radial(p, ds1, h1) \wedge radial(p, ds2, h2) \wedge h1 \neq h2 \rightarrow ds1 \neq ds2, \tag{80}$$

$$position1(pa, dir, p1, pos1) \wedge position1(pa, dir, p2, pos2) \wedge pos1 \neq pos2 \rightarrow p1 \neq p2, \tag{81}$$

$$leftOf(pa, dir, p) \wedge on(pa, q) \rightarrow p \neq q, \tag{82}$$

$$leftOf(pa, dir, p) \wedge on(pa1, p) \rightarrow pa \neq pa1, \tag{83}$$

$$at(ds, p) \wedge at(ds, q) \rightarrow p = q, \tag{84}$$

$$along(ds, pa, dir) \wedge along(ds, pa1, dir1) \rightarrow pa = pa1 \wedge dir = dir1. \tag{85}$$

Rules (77) and (78) rely on the fact that dstates have a unique view and turn actions link different distinctive states (Axioms (22) and (39)). Rule (79) uses the fact that paths are not circular in order to conclude that if p is before q then p and q must be different (Axiom (44)). Rules (80) and (81) use radial and one dimensional frames of reference to conclude inequality of dstates and places, respectively (Axiom (65)). Rules (82) and (83)

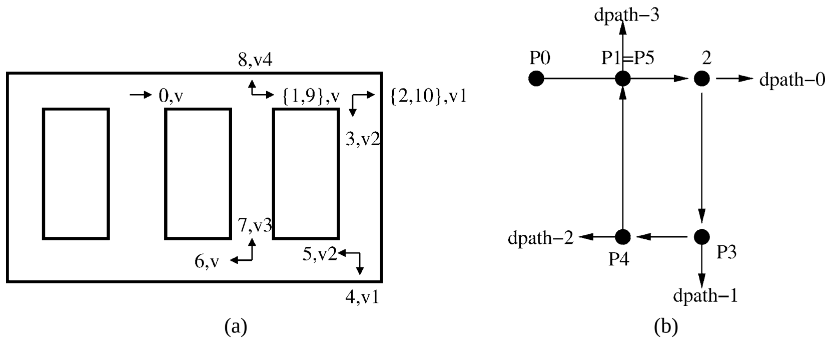

Fig. 24. (a) Numbers identify the dstates created by the map building algorithm. Views associated with dstates are also shown. Dstates 1 and 9 are at the same environment location. (b) Places and dpaths created by the map building algorithm. Notice that $P1$ and $P5$ are two names for the same place.

use boundary relations in order to distinguish places and paths respectively (Axiom (58)).[10] Rules (84) and (85) state that each distinctive state is at a unique place, along a unique path direction (Axioms (29) and (37)).

Assert schema. *Assert-schema*$(pmodel, s)$ creates the places and paths needed to explain s. Instead of asserting $s = \langle ds, a, ds' \rangle$ in *pmodel*, the algorithm asserts $s^* = \langle ds^*, a, ds'^* \rangle$ where ds^* and ds'^* are the representatives in *pmodel* for the *teq* equivalence classes of *ds* and *ds*′. Asserting a schema in Algernon corresponds to creating the frame (object) representing the schema. Forward and backward chaining rules derived from the topological theory are then evaluated, and places and paths needed to explain s are created.

8.2. Trace example

We illustrate the topological map building algorithm with the environment of Fig. 24(a). Distinctive states are visited in the order suggested by the figure. Distinctive state 9 is at the same environment location as dstate 1. However, two topological map are possible: either the agent is back to dstate 1 or dstate 0 (this is Example 13). After traveling from dstate 9 to dstate 10, only one topological map is possible (Fig. 24(b)). Fig. 25 illustrates the use of the topological rules to distinguish distinctive states that share the same view. Fig. 26 shows when branches in the search are created and how they can be refuted as more information becomes available to the agent.[11]

[10] $leftOf(pa, dir, p)$ in the implementation is an abbreviation for Section 6.2's longer expression $leftOf(pa, dir, lr) \wedge in\_region(p, lr)$.

[11] In the implementation, *dpaths* represent ordered dstates linked by travel actions. Dpaths correspond to paths that only have one direction associated with them. Paths are created when the agent has traveled in both direction of a path. At that time, two dpaths are associated with the path, one for each path's direction.

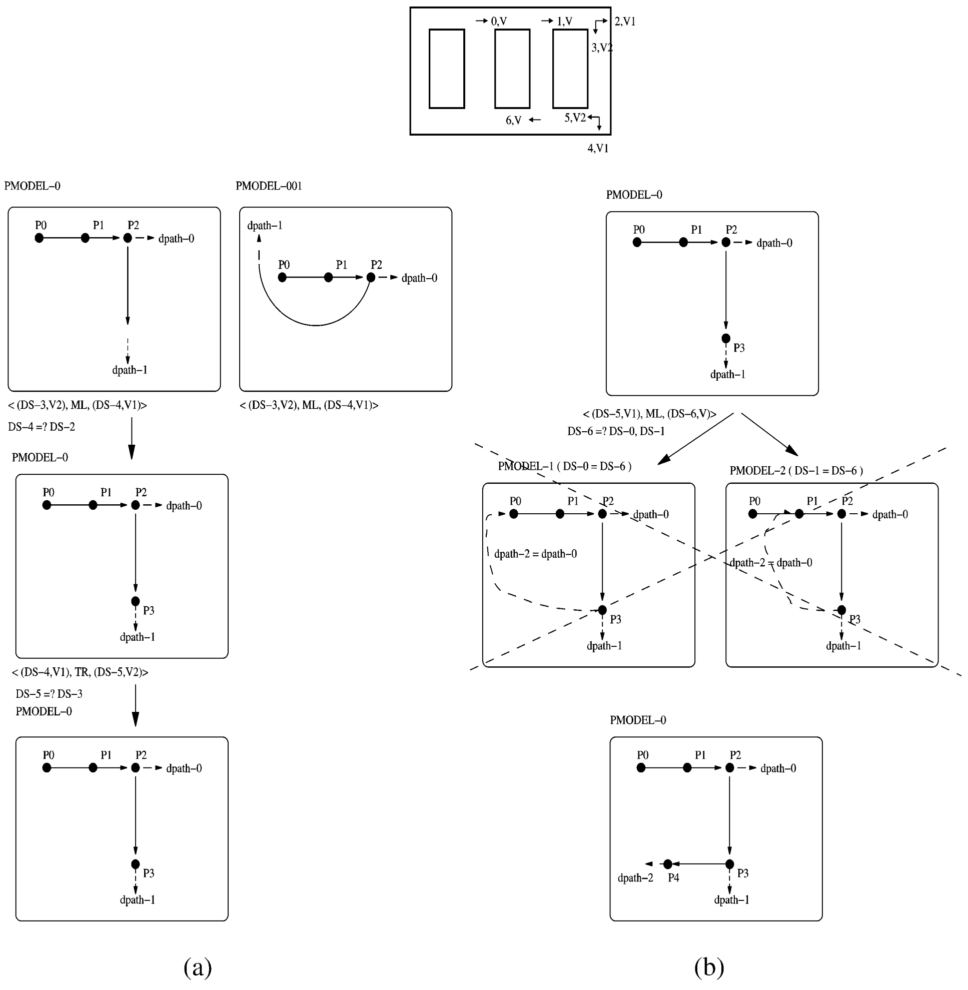

Fig. 25. (a) When the agent reaches dstate 4, the same view $v1$ has been observed at dstate 2. Since dstate 2 is along dpath-0 and dstate 4 will be along dpath-1 (the agent just traveled from dstate 3 along dpath-1), dstate 4 and 2 are proven different. Dpaths 0 and 1 are different since there is a *turnRight* action relating them (Axiom (39)). Place $P3$ is created to be the place dstate 4 is at (Axiom (29)). Place $P3$ is proven different from place $P2$ since $P2$ is before $P3$ along dpath-1 (Corollary 1). Consequently, dstates 5 and 3 are proven different. There are however two possible models depending whether $P3$ is to the right or not of dpath-0. Our boundary regions circumscription policy (Section 6.3) prefers PMODEL-0 in which $P3$ is to the right of dpath-0 over PMODEL-001 in which no boundary relations exist. In this example, the search will never explore further the branch associated with PMODEL-001 because the branch associated with PMODEL-0 leads to a consistent map for the given experiences. (b) The agent travels to dstate 6 along dpath-2. Because $P3$ is to the right of dpath-0, dpath-2 cannot be the same as dpath-0, which makes PMODEL-1 and PMODEL-2 inconsistent. The only remaining (and hence minimal) model is PMODEL-0, in which dstate 6 is different from dstates 0 and 1.

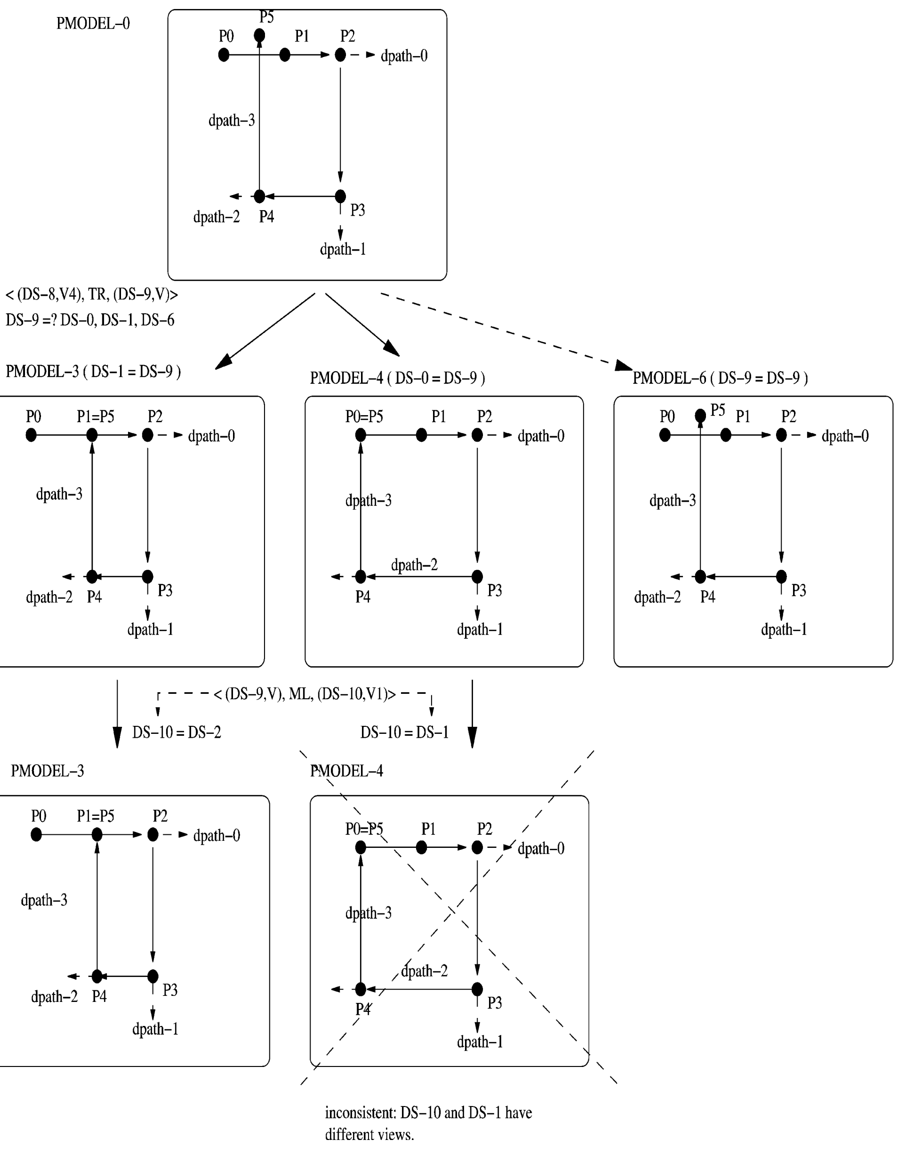

Fig. 26. By the time the agent reaches dstate 8, six places (not five) are part of the map. Places $P5$ and $P1$ are not equal since the dstates are these places are not yet turn related (Axiom (30)). Turning from dstate 8 to 9 leaves the agent with the three possibilities: (pmodel-3) dstates 9 and 1 are equal (and so $P1 = P5$, or (pmodel-4) dstates 9 and 0 are equal (and so $P0 = P1$), or (pmodel-6) dstate 9 is a new different dstate. That dstates 9 and 6 are different follows from the fact that places $P4$ and $P5$ are different. Pmodels 3 and 4 are minimal according to the topological theory circumscription policy. Pmodel-6 is not, but is left as a possible state in the search should new information render the other models inconsistent. The new schema $\langle 9, ML, 10\rangle$ will render pmodel-4 inconsistent. Since actions are deterministic and dstates 0 and 9 are equal in this model, so should dstates 1 and 10. However, these dstates have different views so they cannot be equal. Pmodel-3 will then be the only map associated with the set of experiences.

9. Conclusions

What have we accomplished? We have taken an informal description of the theory of topological maps and provided a formal account of the theory. In addition, we have extended the theory to handle perceptual aliasing, to describe environments with self intersecting and convergent paths, and to deal with local metrical information including uncertainty. The topological theory is independent of the agent's exploration strategy and of the algorithms used to build topological maps. We have taken the theory as a specification for a program able to keep track of different topological maps consistent with the agent's experiences in the environment. This program supports different exploration strategies as well as facilitates map disambiguation when the case arises.

A logical account of the causal, topological and local metrical theories was given using Nested Abnormality Theories. The minimality conditions embedded in the formalization define the preferred models associated with the theories. In Sections 4–7 we illustrated the main properties of the theories. In particular we showed how the minimal models associated with these theories are adequate models for the spatial knowledge an agent has about its environment. We also demonstrated how the causal, topological, and local metrical levels of the representation assume different spatial properties of the actions performed by the agent. This provides an increasingly refined ability to infer or refute equality relations (*ceq* and *teq*) among experienced environment states. By clarifying the ontology of causal and topological maps, and determining the dependency structure of the non-monotonic theory, we provide a solid foundation for general-purpose strategies for exploring unknown environments, or for disambiguating cases of perceptual aliasing.

The circumscription priority ordering embedded in the theory is a result of our research, as we experimented with various orders to determine which ordering defined models that corresponded to what is intuitively the "correct map" of the environment. Because we have no formal definition of what the correct map is, it is impossible to prove mathematically that the circumscription priority ordering is the correct one. Possibly future research can provide such a formal definition, but the difficulties arise from handling partial experience in the environment, or highly symmetrical environments with a great deal of perceptual aliasing.

How useful is this theory? This work defines topological maps independently of the algorithms used to create such maps. The theory is general in that it covers the major ideas in the field of spatial representation using topological maps. The theory is useful in that it specifies the minimal set of objects and relationships any topological map building implementation should have. Although our theory covers most of the known ideas about topological maps, it is not just a union of previous work in a common framework. The theory defines different spatial ontologies (causal, topological, metrical), illustrates what spatial knowledge is captured by each ontology, and then shows the relationships among these ontologies. The theory shows how the combined spatial knowledge associated with the different ontologies results in a different "map" from the one associated with each independent ontology.

The axiomatic theory has practical value. It has been used to build cognitive maps by both physical and simulated robots. [24,29,49] explicitly use the axiomatic theory described in this paper as well as the implemented algorithms in order to build topological maps. [49] shows how a wheelchair robot builds the topological map of a building's floor. The major focus of this work was on testing the applicability and correctness of the axioms and algorithms here described.

In [24] topological maps are built as a mean to disambiguate distinctive states with the same view. The map provides an unambiguous assignment of distinctive states to views, which can then used by the robot to "refine its views" so that it is possible to distinguish distinctive states from sensory information alone. "Lassie (the robot)... collected 240 images from 20 distinctive states. The topological map linking them contained seven places and four paths.... By building the causal and topological map the robot is able to disambiguate all twenty distinctive states, even though there are only ten different views..." [24].

Finally, Fig. 9 in [29] describes experimental results where a simulated agent builds a topological map and learns boundary relations for grid-like environments. This work presents a computational hypothesis that describes how the "skeleton" of major paths emerges from the interaction of three factors: "(i) the topological map is represented as a bipartite graph of places and paths, where a path is a one-dimensional ordered set of places, (ii) a traveler incrementally accumulates topological relationships, including the relation of a place to a path serving as a dividing boundary separating two regions; and (iii) the wayfinding algorithm prefers paths rich in boundary relations so they are likely to acquire more boundary relations. This positive-feedback loop leads to an oligarchy of paths rich in boundary relations (i.e., the skeleton in the cognitive map)" [29].

Acknowledgements

We are grateful to Vladimir Lifschitz for his valuable feedback during this work. We also thanks the anonymous referees for their helpful questions and suggestions.

Appendix A. Nested abnormality theories

In this appendix we define circumscription and nested abnormalities theories following [35,36]. The main idea of circumscription is to consider, instead of arbitrary models of an axiom set, only the models that satisfy a certain minimality condition (usually set inclusion).

Definition A.1 (*Circumscription*). Let $A(P, Z_1, \ldots, Z_m)$ be a sentence containing a predicate constant P and object, function and/or predicate constants $Z_1, \ldots, Z_m$ (and possibly other object, function and predicate constants). The *circumscription of P in A with varied $Z_1, \ldots, Z_m$* is the sentence

$$A(P, Z_1, \ldots, Z_m) \wedge \neg\exists p, z_1, \ldots, z_m \big[A(p, z_1, \ldots, z_m) \wedge p < P\big] \tag{A.1}$$

where $p < P$ denotes the formula

$$\forall x \quad \big\{p(x) \to P(x)\big\} \wedge \exists x \big\{\neg p(x) \wedge P(x)\big\}.$$

We denote formula (A.1) by $\mathit{CIRC}[A; P; Z]$.

Intuitively, the models of $\mathit{CIRC}[A; P; Z]$ are the models of A in which the extent of P cannot be smaller without losing the property A, even at the price of changing the interpretations of the constants Z.

It is often convenient to arrange different defaults by assigning priorities to them. Next we define two extensions to the basic definition of circumscription: parallel and prioritized circumscription.

Definition A.2 (*Parallel circumscription*). The *parallel circumscription*

$$\mathit{CIRC}\big[A; P^1, \ldots, P^n; Z\big]$$

is the sentence $A(P, Z) \wedge \neg\exists p, z[A(p, z) \wedge p \prec P]$, where P stands for the tuple of predicates $P^1, \ldots, P^n$ and $p \prec P$ stands for the formula $\forall 1 \leqslant i \leqslant n\ p^i \leqslant P^i \wedge \exists 1 \leqslant i \leqslant n\ p^i < P^i$.

Definition A.3 (*Prioritized circumscription*). The *prioritized circumscription*

$$\mathit{CIRC}\big[A; P^1 \succ \cdots \succ P^n; Z\big]$$

is the sentence $A(P, Z) \wedge \neg\exists p, z[A(p, z) \wedge p \prec P]$, where P stands for the tuple of predicates $P^1, \ldots, P^n$ and $p \prec P$ stands for the formula

$$\bigvee_{i=1}^{n}\left(\bigwedge_{j=1}^{i-1}\big(p^j = P^j\big) \wedge \big(p^i < P^i\big)\right).$$

The formula $p \prec P$ defines a *lexicographic* order among the predicates in p and P. Proposition 15 in [35] shows that prioritized circumscription can be reduced to parallel circumscription as follows:

Theorem A.1. *The circumscription* $\mathit{CIRC}[A; P^1 \succ \cdots \succ P^n; Z]$ *is equivalent to*

$$\bigwedge_{i=1}^{n} \mathit{CIRC}\big[A; P^i; P^{i+1}, \ldots, P^n, Z\big].$$

Notation A.1. $\mathit{CIRC}[A; P^1 \succ \ldots \neg P_i \ldots \succ P^n; Z]$ stands for the formula

$$\mathit{CIRC}\big[A \wedge \mathit{not\_P}_i \equiv \neg P_i; P^1 \succ \ldots \mathit{not\_P}_i \ldots \succ P^n; Z, P_i\big]$$

where $\mathit{not\_P}_i$ is a new constant predicate not occurring in A.

A.1. Nested abnormality theories (NATs)

Nested abnormality theories allows one to apply the circumscription operator to a subset of axioms, by structuring the knowledge base (the theory) into blocks. Each block can be viewed as a group of axioms that describes a certain collection of predicates and functions, and the nesting of blocks reflects the dependence of these descriptions on each other.

Definition A.4 (*NATs*). Consider a second-order language L that does *not* include *Ab* among its symbols. For every natural number k, by L_k we denote the language obtained from L by adding *Ab* as a k-ary predicate constant. *Blocks* are defined recursively as follows: For any k and any list of function and/or predicate constants $C_1, \dots, C_m$ of L, if each of $A_1, \dots, A_n$ is a formula of L_k or a *block*, then $\{C_1, \dots, C_m\colon\ A_1, \dots, A_n\}$ is a *block*. The last expression reads: $C_1, \dots, C_m$ are such that $A_1, \dots, A_n$. About $C_1, \dots, C_m$ we say that they are *described* by this block.

The semantics of NATs is characterized by a map φ that translates blocks into sentences of L. It is convenient to make φ defined also on formulas of the languages L_k. If A is such a formula, then $\varphi(A)$ stands for the universal closure of A. For blocks we define, recursively:

$$\varphi\{C_1, \dots, C_m : A_1, \dots, A_n\} = \exists ab\, \mathit{CIRC}[\varphi A_1, \dots, \varphi A_n : ab : C_1, \dots, C_m].$$

Most often, it is desirable not to mention the predicate *Ab* at all. We will adopt the following notations:

- $\{C_1, \dots, C_m, \mathit{min}\, P : A_1, \dots, A_n\}$ stands for

$$\{C_1, \dots, C_m, P : P(x) \to Ab(x),\ A_1, \dots, A_n\}$$

- $\{C_1, \dots, C_m, \mathit{max}\, P : A_1, \dots, A_n\}$ stands for

$$\{C_1, \dots, C_m, P : \neg Ab(x) \to P(x),\ A_1, \dots, A_n\}.$$

Definition A.5. We extend the definition of *blocks* as follows: if A is a block, so is $\mathit{CIRC}[A;\, P^1 \succ \cdots \succ P^n;\, Z]$. The semantics of NATs is extended such that

$$\phi \mathit{CIRC}\big[A;\, P^1 \succ \cdots \succ P^n;\, Z\big] = \mathit{CIRC}\big[\phi A;\, P^1 \succ \cdots \succ P^n;\, Z\big].$$

As the next theorem shows, in some cases prioritized circumscription can be expressed using NAT's. In these cases however, the notation for prioritized circumscription is more compact than its equivalent NAT's. This motivates our previous definition.

Theorem A.2. *Let A be a sentence such that Ab does not occur in A. Then,*

$$\mathit{CIRC}[A;\, P \succ Q;\, Z] = \{Z,\ \mathit{min}\, Q : \{Z,\ Q, \mathit{min} P\colon\ A\}\}.$$

Appendix B. ceq properties

In this appendix we provide proofs for the different properties of the predicated *ceq* defined in Section 4.

Theorem 1. *Let E be a complete set of experiences and let CEQ_block be defined as follows*:

$\{\mathit{max}\, \mathit{ceq}$:

$\mathit{ceq}(ds_1, ds_2) \to \mathit{View}(ds_1, v) \equiv \mathit{View}(ds_2, v)$,

$$ceq(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \rightarrow ceq(ds_1', ds_2')$$
}

Then the predicate *ceq* is an equivalence relation.

Proof. Let M_1 be a model for the axioms inside the *CEQ_block* as well as the other axioms of $CT(E)$. Let M_2 be a structure identical to M_1 except that

$$ceq^{M_2}(ds, ds') \equiv ceq^{M_1}(ds, ds') \vee ds = ds'.$$

We are to prove that M_2 is a model for the axioms inside the *CEQ_block* and consequently *CEQ_block* $\models ceq(ds, ds)$.[12] Indeed,

- $M_2 \models ceq(ds, ds') \rightarrow ceq(ds', ds)$. In fact,

$$\begin{aligned} ceq^{M_2}(ds, ds') &\equiv ceq^{M_1}(ds, ds') \vee ds = ds' \\ &\rightarrow ceq^{M_1}(ds', ds) \vee ds' = ds \\ &\equiv ceq^{M_2}(ds', ds). \end{aligned}$$

- $M_2 \models ceq(ds, ds') \wedge ceq(ds', ds'') \rightarrow ceq(ds, ds'')$. In fact,

$$\begin{aligned} &ceq^{M_2}(ds, ds') \wedge ceq^{M_2}(ds', ds'') \\ &\quad\equiv \big(ceq^{M_1}(ds, ds') \vee ds = ds'\big) \wedge \big(ceq^{M_1}(ds', ds'') \vee ds' = ds''\big) \\ &\quad\equiv \big(ceq^{M_1}(ds, ds') \wedge ceq^{M_1}(ds', ds'')\big) \vee \big(ds = ds' \wedge ceq^{M_1}(ds', ds'')\big) \vee \\ &\qquad \big(ceq^{M_1}(ds, ds') \wedge ds' = ds''\big) \vee (ds = ds' \wedge ds' = ds'') \\ &\quad\rightarrow ceq^{M_1}(ds, ds'') \vee (ds = ds' \wedge ds' = ds'') \\ &\quad\equiv ceq^{M_2}(ds, ds''). \end{aligned}$$

- $M_2 \models ceq(ds, ds') \rightarrow View(ds, v) \equiv View(ds', v)$. In fact,

$$\begin{aligned} ceq^{M_2}(ds, ds') &\equiv ceq^{M_1}(ds, ds') \vee ds = ds' \\ &\rightarrow \forall v\big[View(ds, v) \equiv View(ds', v)\big] \vee ds = ds' \\ &\rightarrow \forall v\big[View(ds, v) \equiv View(ds', v)\big] \vee \forall v\big[View(ds, v) \equiv View(ds', v)\big] \\ &\equiv View(ds, v) \equiv View(ds', v). \end{aligned}$$

- $M_2 \models ceq(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \rightarrow ceq(ds_1', ds_2'')$. In fact,

$$\begin{aligned} &ceq^{M_2}(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \\ &\quad\equiv \big(ceq^{M_1}(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle\big) \vee \\ &\qquad \big(ds_1 = ds_2 \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle\big) \\ &\quad\rightarrow ceq^{M_1}(ds_1', ds_2') \vee \big(\langle ds_1, a, ds_1'\rangle \wedge \langle ds_1, a, ds_2'\rangle\big) \end{aligned}$$

[12] M_2 satisfies the other axioms in $CT(E)$ since *ceq* does not occur in them.

$$\begin{aligned}&\overset{(15)}{\to} ceq^{M_1}(ds'_1, ds'_2) \vee ds'_1 = ds'_2\\&\equiv ceq^{M_2}(ds'_1, ds'_2).\end{aligned}$$

Let us prove that *CEQ_block* $\models ceq(ds, ds') \to ceq(ds', ds)$. Let M_2 be a model identical to M_1 except that

$$ceq^{M_2}(ds, ds') = ceq^{M_1}(ds, ds') \vee ceq^{M_1}(ds', ds).$$

By definition, ceq^{M_2} is symmetric. We need to prove that M_2 satisfy the axioms inside *CEQ_block*:

- $M_2 \models ceq(ds, ds') \to View(ds, v) \equiv View(ds', v)$. In fact,

$$\begin{aligned}&ceq^{M_2}(ds, ds') \equiv ceq^{M_1}(ds, ds') \vee ceq^{M_1}(ds', ds)\\&\quad\to \forall v\big[View(ds, v) \equiv View(ds', v)\big] \vee \forall v\big[View(ds', v) \equiv View(ds, v)\big]\\&\quad\equiv View(ds, v) \equiv View(ds', v).\end{aligned}$$

- $M_2 \models ceq(ds_1, ds_2) \wedge \langle ds_1, a, ds'_1\rangle \wedge \langle ds_2, a, ds'_2\rangle \to ceq(ds'_1, ds'_2)$. In fact,

$$\begin{aligned}&ceq^{M_2}(ds_1, ds_2) \wedge \langle ds_1, a, ds'_1\rangle \wedge \langle ds_2, a, ds'_2\rangle\\&\quad\equiv \big[ceq^{M_1}(ds_1, ds_2) \wedge \langle ds_1, a, ds'_1\rangle \wedge \langle ds_2, a, ds'_2\rangle\big] \vee\\&\qquad\big[ceq^{M_1}(ds_2, ds_1) \wedge \langle ds_1, a, ds'_1\rangle \wedge \langle ds_2, a, ds'_2\rangle\big]\\&\quad\to ceq^{M_1}(ds'_1, ds'_2) \vee ceq^{M_1}(ds'_2, ds'_1)\\&\quad\equiv ceq^{M_2}(ds'_1, ds'_2).\end{aligned}$$

Finally, let us prove that *CEQ_block* $\models ceq(ds, ds') \wedge ceq(ds', ds'') \to ceq(ds, ds'')$. Let M_2 be a model identical to M_1 except that

$$ceq^{M_2} = transitive\_closure(ceq^{M_1}).$$

By definition, ceq^{M_2} is transitive. If ceq^{M_1} is reflexive and symmetric, so is ceq^{M_2}. We need to prove that M_2 satisfies the axioms inside *CEQ_block*:

- $M_2 \models ceq(ds, ds') \to View(ds, v) \equiv View(ds', v)$. In fact,

$$\begin{aligned}&ceq^{M_2}(ds, ds')\\&\quad\equiv \exists ds^0, ds^1, \ldots, ds^n\big[ds = ds^0,\ ds' = ds^n,\ ceq^{M_1}(ds^i, ds^{i+1}),\ 0 \leqslant i < n\big]\\&\quad\to \exists ds^0, ds^1, \ldots, ds^n\\&\qquad\big[ds = ds^0,\ ds' = ds^n,\ View(ds^i, v) \equiv View(ds^{i+1}, v),\ 0 \leqslant i < n\big]\\&\quad\to \exists ds^0, ds^n\big[ds = ds^0,\ ds' = ds^n, View(ds^0, v) \equiv View(ds^n, v)\big]\\&\quad\equiv View(ds, v) \equiv View(ds', v).\end{aligned}$$

- $M_2 \models ceq(ds_1, ds_2) \wedge \langle ds_1, a, ds'_1\rangle \wedge \langle ds_2, a, ds'_2\rangle \to ceq(ds'_1, ds'_2)$. In fact,

$$\begin{aligned}
&ceq^{M_2}(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \\
&\quad \equiv \exists ds^i (1 \leqslant i \leqslant n)\big[ds_1 = ds^1,\ ds_2 = ds^n,\ ceq^{M_1}(ds^i, ds^{i+1}),\ 1 \leqslant i < n\big] \wedge \\
&\qquad \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \\
&\quad \overset{hyp.}{\rightarrow} \exists ds^i \exists \langle ds^i, a, ds^{i'}\rangle \\
&\qquad \big[ds_1 = ds^1, ds_2 = ds^n, ds_1' = ds^{1'}, ds_2' = ds^{n'}, ceq^{M_1}(ds^i, ds^{i+1}), \\
&\qquad 1 \leqslant i < n\big] \\
&\quad \rightarrow \exists ds^{i'}\big[ds_1' = ds^{1'},\ ds_2' = ds^{n'},\ ceq^{M_1}(ds^{i'}, ds^{(i+1)'}),\ 1 \leqslant i < n\big] \\
&\quad \equiv ceq^{M_2}(ds_1', ds_2'). \qquad \square
\end{aligned}$$

When a set of experiences is complete the predicate *ceq* captures the idea that two distinctive states are the same if they render the same views under any sequence of actions. Assume that E is complete and let $A = a_1, \ldots, a_n$ denote a sequence of actions. The term $A(ds)$ denotes the distinctive state resulting from executing A starting at ds. By definition, $A(ds) = ds$ if $n = 0$, $A(ds) = ds'$ such that $E \models \langle \langle a_1, \ldots, a_{n-1}\rangle(ds), a_n, ds'\rangle$. Notice that the definition of $A(ds)$ makes sense since E is complete and actions are deterministic.

Theorem 2. *Let E be a complete set of experiences. Then,*

$$ceq(ds, ds') \equiv \forall A, v\big[View(A(ds), v) \equiv View(A(ds'), v)\big].$$

Proof. Let M_1 be a model for the axioms inside the *CEQ_block* as well as the other axioms of $CT(E)$. Let M_2 be a model identical to M_1 except that

$$ceq^{M_2}(ds, ds') \equiv \forall A, v\,\big[View(A(ds), v) \equiv View(A(ds'), v)\big].$$

By induction in the length of action sequences on can prove that $ceq^{M_1} \subseteq ceq^{M_2}$. Our proof is complete by showing that M_2 satisfies the axioms inside *CEQ_block*:

- $M_2 \models ceq(ds, ds') \rightarrow View(ds, v) \equiv View(ds', v)$. In fact, suppose $M_2 \models ceq(ds, ds')$ and consider the empty sequence of actions, $A = \{\}$, $A(ds) = ds$. Then

$$View(ds, V) \equiv View(A(ds), v) \equiv View(A(ds'), v) \equiv View(ds', v).$$

- $M_2 \models ceq(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \rightarrow ceq(ds_1', ds_2')$. In fact,

$$\begin{aligned}
ceq^{M_2}(ds_1', ds_2') &\equiv \forall A, v\,\big[View(A(ds_1'), v) \equiv view(A(ds_2'), v)\big] \\
&\leftarrow \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle \wedge \\
&\qquad \forall A, v\,\big[View(aA(ds_1), v) \equiv View(aA(ds_2), v)\big] \\
&\leftarrow ceq^{M_2}(ds_1, ds_2) \wedge \langle ds_1, a, ds_1'\rangle \wedge \langle ds_2, a, ds_2'\rangle. \qquad \square
\end{aligned}$$

Appendix C. teq properties

In this appendix we prove some properties of the SSH topological theory. Recall the SSH topological theory is defined as follows:

$$
\begin{aligned}
&TT(E) = \\
&\quad \textit{there exist infinitely many places}, \\
&\quad \textit{there exist infinitely many paths}, \\
&\quad \neg\exists p\big[\mathit{tplace}(p) \wedge \mathit{is\_region}(p)\big], \\
&\quad \neg\exists pa\big[\mathit{tpath}(pa) \wedge \mathit{route}(pa)\big], \\
&\quad \mathit{COMPLETION}(E), \\
&\quad \text{Axioms (2)–(10)}, \\
&\quad \langle ds, a, ds'\rangle \wedge \langle ds, a, ds''\rangle \rightarrow ds' = ds'', \quad \text{(Axiom (15))} \\
&\quad T\_block, \\
&\quad\; AT\_block = && \text{(C.1)} \\
&\quad\;\; \{\, \mathit{max}\, \mathit{teq}: \\
&\quad\;\;\; \Gamma \\
&\quad\;\;\; \mathbf{circ}\, \mathit{tpath} \succ \mathit{tplace}\, \mathbf{var}\, \overrightarrow{\mathit{SSHpred}} && \text{(C.2)} \\
&\quad\;\; \}
\end{aligned}
$$

where Γ is the set of axioms defined on Block (20) (Section 5.2), and $\overrightarrow{\mathit{SSHpred}}$ stands for the tuple of predicates $\langle$**at**, **along**, **order**, **on**, **teq**, *turn_eq*, *travel_eq*$\rangle$.

Proposition C.1. *Let M be a model of $TT(E)$. Then,*

- $M \models \forall pa,\ [\mathit{tpath}(pa) \equiv \exists ds, dir\, \mathit{along}(ds, pa, dir)]$.
- $M \models \forall p,\ [\mathit{tplace}(p) \equiv \exists ds\, \mathit{at}(ds, p)]$.

Proof.

$$
\begin{aligned}
&\mathit{CIRC}[\Gamma; \mathit{tpath} \succ \mathit{tplace}; \mathit{SSHpred}] \\
&\equiv \qquad \{\text{Proposition 15 in [35]}\} \\
&\mathit{CIRC}[\Gamma; \mathit{tpath}; \mathit{tplace}, \mathit{SSHpred}] \wedge \mathit{CIRC}[\Gamma; \mathit{tpath}, \mathit{tplace}; \mathit{SSHpred}] \\
&\rightarrow \qquad \{\text{def. of circumscription}\} \\
&\mathit{CIRC}[\Gamma; \mathit{tpath}].
\end{aligned}
$$

Since $\Gamma = \Gamma'(\mathit{tpath}) \wedge [\mathit{along}(ds, pa, dir) \rightarrow \mathit{tpath}(pa)]$ where $\Gamma'(\mathit{tpath})$ is negative, then

$$
\begin{aligned}
& \mathit{CIRC}[\Gamma; \mathit{tpath}] \\
& \equiv \\
& \mathit{CIRC}\big[\Gamma'(\mathit{tpath}) \wedge [\mathit{along}(ds, pa, dir) \to \mathit{tpath}(pa)]; \mathit{tpath}\big] \\
& \equiv \qquad \{\text{Proposition 4 in [35]}\} \\
& \Gamma'(\mathit{tpath}) \wedge \mathit{CIRC}\big[\mathit{along}(ds, pa, dir) \to \mathit{tpath}(pa); \mathit{tpath}\big] \\
& \to \qquad \{\text{Proposition 1 in [35]}\} \\
& \big[\exists ds, dir\, \mathit{along}(ds, pa, dir)\big] \equiv \mathit{tpath}(pa).
\end{aligned}
$$

Similarly, $\Gamma = \Gamma' \wedge [at(ds, p) \to tplace(p)]$ where *tpath* does not occur in Γ'. Then,

$$
\begin{aligned}
& \mathit{CIRC}[\Gamma; \mathit{tpath} \succ \mathit{tplace}; \mathit{SSHpred}] \\
& \to \qquad \{\text{see above}\} \\
& \mathit{CIRC}[\Gamma; \mathit{tpath}, \mathit{tplace}; \mathit{SSHpred}] \\
& \to \qquad \{\text{def. parallel circumscription}\} \\
& \mathit{CIRC}[\Gamma; \mathit{tpath}, \mathit{tplace}] \\
& \to \qquad \{\text{def. parallel circumscription}\} \\
& \mathit{CIRC}\big[\Gamma' \wedge [at(ds, p) \to \mathit{tplace}(p)]; \mathit{tplace}\big] \\
& \equiv \qquad \{\text{Propositions 1 and 4 in [35]}\} \\
& \Gamma' \wedge \big[\exists ds, at(ds, p)\big] \equiv \mathit{tplace}(p). \qquad \square
\end{aligned}
$$

Proposition C.2. *The topological map associated with a finite set of experiences E has a finite number of topological paths and a finite number of topological places.*

Proof. Since a distinctive state is along at most one topological path (Axiom (37)), Proposition C.1 implies that for any model M of $TT(E)$ there is an injection from tpath^M into $\mathit{distinctive\text{-}states}^M$. Since $\mathit{distinctive\text{-}states}^M$ is finite so is tpath^M.

Similarly, since distinctive states are at a unique topological place (Axiom (29)), from Proposition C.1 we conclude that the set of topological places in a model of $TT(E)$ is finite. □

Theorem 3. *Let ds_1 be a distinctive state symbol such that*

$$\forall ds_2 \notin [ds_1]_{\widehat{turn}},\ [ds_2]_{teq} \cap [ds_1]_{\widehat{turn}} = \emptyset. \tag{C.3}$$

Then

$$\forall ds_2 \notin [ds_1]_{\widehat{turn}},\ place(ds_2) \neq place(ds_1).$$

Proof. The hypothesis of the theorem implies that

$$\forall ds_2 \notin [ds_1]_{\widehat{turn}},\ \neg turn\_eq(ds_2, ds_1).$$

Indeed,

$$\begin{aligned} turn\_eq(ds_1, ds_2) \equiv \exists b_0, \ldots, b_n, b_{0'}, \ldots, b_{n'}\ s.t.\ & \\ & \bullet\, b_0 = ds_2, \quad b_{n'} = ds_1, \\ & \bullet\, teq(b_i, b_{i'}), \quad i = 0, \ldots, n \\ & \bullet\, \widehat{turn}(b_{i'}, b_{i+1}), \quad i = 0, \ldots, n-1\,. \end{aligned}$$

Let $1 \leqslant j \leqslant n$ such that $[\forall j \leqslant k \leqslant n,\ b_{k'} \in [ds_1]_{\widehat{turn}}]$ and $b_{(j-1)'} \notin [ds_1]_{\widehat{turn}}$. Notice that such a j exists since $ds_1 = b_{0'} \notin [ds_1]_{\widehat{turn}}$ and $ds_1 = b_{n'} \in [ds_1]_{\widehat{turn}}$. Consequently,

$$\begin{aligned} & turn\_eq(ds_1, ds_2) \\ & \rightarrow \\ & b_{j'} \in [ds_1]_{\widehat{turn}} \\ & \rightarrow \qquad \{teq(b_j, b_{j'})\} \\ & [b_j]_{teq} \cap [ds_1]_{\widehat{turn}} \neq \emptyset \\ & \rightarrow \qquad \{\text{C.3}\} \\ & b_j \in [ds_1]_{\widehat{turn}} \\ & \rightarrow \qquad \{\widehat{turn}(b_{(j-1)'}, b_j)\} \\ & b_{(j-1)'} \in [ds_1]_{\widehat{turn}} \\ & \rightarrow \\ & false. \end{aligned}$$

Thus $\neg turn\_eq(ds_2, ds_1)$ should be the case. □

Theorem 4. *Any two models of the SSH topological theory have the same number of topological paths and the same number of topological places.*

Proof. In order to prove that two models M_1 and M_2 of $TT(E)$ have the same number of topological paths (tpaths) and the same number of topological places (tplaces), it is enough to show that this is the case for models of the *AT_block* (Block (C.1)). Suppose that $tpath^{M_1}$ has less elements than $tpath^{M_2}$, and so there exists an injection $\phi : tpath^{M_1} \rightarrow tpath^{M_2}$. *One can extend ϕ to define an isomorphism from M_1 into M_2', such that $M_2' \prec M_2$, where $\prec$ is the order defined by the circumscription policy* C.2. This proves that M_1 and M_2 have the same number of topological paths. In fact,

- Let $\phi : tplace^{M_1} \rightarrow places^{M_2}$ be an injection. Such an injection exists since $tplace^{M_1}$ is finite and $places^{M_2}$ is infinite.
- Let $\phi : S^{M_1} \rightarrow S^{M_2}$ be the identity over the sorts (S) of distinctive states, actions, views, schemas, path types and path directions. Recall we assumed a *Herbrand* interpretation for these sorts, where the corresponding universes are defined by the constant symbols in E.

The function ϕ above defines an isomorphic embedding from M_1 into M_2 in the standard way. In fact, $\phi(M_1) = M_2'$ is defined as follows:

- $tpath^{M_2'} = \phi(tpath^{M_1})$, $tplace^{M_2'} = \phi(tplace^{M_1})$.
- $teq^{M_2'} = \phi(teq^{M_1}) = \{teq(ds_1, ds_2)\colon\ M_1 \models teq(ds_1, ds_2)\} = teq^{M_1}$.
- $at^{M_2'} = \phi(at^{M_1}) = \{at(ds, \phi(p))\colon\ M_1 \models at(ds, p)\}$.
- $along^{M_2'} = \phi(along^{M_1}) = \{along(ds, \phi(pa), dir)\colon\ M_1 \models along(ds, pa, dir)\}$.
- $order^{M_2'} = \phi(order^{M_1}) = \{order(\phi(pa), dir, \phi(p), \phi(q))\colon\ M_1 \models order(pa, dir, p, q)\}$.
- $on^{M_2'} = \phi(on^{M_1}) = \{on(\phi(pa), \phi(p))\colon\ M_1 \models on(pa, p)\}$.
- $turn\_eq^{M_2'} = \phi(turn\_eq^{M_1}) = turn\_eq^{M1}$.
- $travel\_eq^{M_2'} = \phi(travel\_eq^{M_1}) = travel\_eq^{M1}$.

Notice that the language of Γ is defined by $\{tpath, tplace\} \cup SSHpred$. Thus $M_1 \models \Gamma$ implies $\phi(M_1) \models \Gamma$. Notice that the circumscription policy varies all predicates in the language of Γ, and ϕ is the identity over all constant symbols in the theory, for otherwise, $\phi(M_1) \models \Gamma$ is not necessarily the case. In general the interpretations of an unary predicate (set) under a circumscriptive theory do not have the same number of elements. For example, consider the models of $CIRC[(P(0) \wedge P(1)) \vee P(2); P]$, where the interpretation of P could have one or two elements (this example is due to Vladimir Lifschitz).

Since $\phi(tpath^{M_1}) \subset tpath^{M_2}$, then $\phi(M_1) \prec M_2$, and so M_2 is not minimal, and is therefore not a model of $TT(E)$. It follows that M_1 and M_2 have the same number of topological paths.

Similar argument shows that M_1 and M_2 have the same number of topological places. If not, there would exists $\phi : tpath^{M_1} \to tpath^{M_2}$ a bijection and $\phi : tplace^{M_1} \to tplace^{M_2}$ an injection that allows us to apply the same argument as above. $\square$

Appendix D. Theory axioms

The block **T_block** inside Block (19) in Section 5.2 defines the properties of the predicates $\widehat{turn}$, $\widehat{travel}$, and $\overrightarrow{travel}$. $\widehat{turn}$ is the equivalence closure of the schemas $\langle \cdot, turn, \cdot \rangle$; $\widehat{travel}$ and $\overrightarrow{travel}$ are the equivalence and transitive closure of the schemas $\langle \cdot, travel, \cdot \rangle$ respectively.[13]

$$
\begin{aligned}
T\_block = \{\ & min\, \widehat{turn}, min\, \widehat{travel}, min\, \overrightarrow{travel}\colon \\
& \langle ds, turn, ds' \rangle \to \widehat{turn}(ds, ds'), \\
& \langle ds, travel, ds' \rangle \to \widehat{travel}(ds, ds') \wedge \overrightarrow{travel}(ds, ds'), \\
& \\
& \widehat{turn}(ds, ds), \\
& \widehat{turn}(ds, ds') \to \widehat{turn}(ds', ds), \\
& \widehat{turn}(ds, ds') \wedge \widehat{turn}(ds', ds'') \to \widehat{turn}(ds, ds''),
\end{aligned}
$$

[13] A block of the form $\{C_1, \ldots, C_n,\ min\, P_1, \ldots, min\, P_k\colon\ A_1, \ldots, A_m\}$ denotes the set of blocks $\{C_1, \ldots, C_n,\ min\, P_1\colon\ A_1, \ldots, A_m\}, \ldots, \{C_1, \ldots, C_n,\ min\, P_k\colon\ A_1, \ldots, A_m\}$.

$$\widehat{travel}(ds, ds),$$
$$\widehat{travel}(ds, ds') \to \widehat{travel}(ds', ds),$$
$$\widehat{travel}(ds, ds') \wedge \widehat{travel}(ds', dr) \to \widehat{travel}(ds, dr),$$

$$\vec{travel}(ds, ds') \wedge \vec{travel}(ds', ds'') \to \vec{travel}(ds, ds'')$$

}

References

[1] D. Angluin, On the complexity of minimum inference of regular sets, Inform. and Control 39 (1978) 337–350.

[2] K. Basye, T. Dean, L.P. Kaelbling, Learning dynamics: System identification for perceptually challenged agents, Artificial Intelligence 72 (1) (1995) 139–171.

[3] A.F. Beardon, Complex Analysis, Wiley, New York, 1979.

[4] J. Borenstein, Y. Koren, The vector field histogram—Fast obstacle-avoidance for mobile robots, IEEE J. Robotics and Automation 7 (3) (1991) 278–288.

[5] J. Borenstein, H.R. Everett, L. Feng, Navigating Mobile Robots: Systems and Techniques, A.K. Peters, Wellesley, MA, 1996.

[6] I. Borg, P. Groenen, Modern Multidimensional Scaling: Theory and Applications, Springer, New York, 1997.

[7] H. Choset, K. Nagatani, Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization, IEEE Trans. on Robotics and Automation 17 (2) (2001) 125–137.

[8] J. Crawford, B. Kuipers, Algernon: A tractable system for knowledge representation, SIGART Bull. 2 (3) (1991) 35–44.

[9] E. Davis, The MERCATOR representation of spatial knowledge, in: A. Bundy (Ed.), Proc. IJCAI-83, Karlsruhe, Germany, Morgan Kaufmann, Los Altos, CA, 1983, pp. 295–301.

[10] T. Dean, K. Basye, L. Kaelbling, Uncertainty in graph-based map learning, in: J.H. Connell, S. Mahadevan (Eds.), Robot Learning, Kluwer Academic, Dordrecht, 1993, pp. 171–192.

[11] G. Dudek, M. Jenkin, E. Milios, D. Wilkes, Robotic exploration as graph construction, IEEE Trans. on Robotics and Automation 7 (6) (1991) 859–865.

[12] H.F. Durrant-Whyte, Consistent integration and propagation of disparate sensor observations, Internat. J. Robotics Res. 6 (3) (1987) 3–24.

[13] H.F. Durrant-Whyte, Integration, Coordination and Control of Multisensor Robot Systems, Kluwer Academic, Boston, MA, 1988.

[14] H.F. Durrant-Whyte, Uncertain geometry in robotics, IEEE J. Robotics and Automation 5 (6) (1988) 23–31.

[15] A. Elfes, Sonar-based real-world mapping and navigation, IEEE J. Robotics and Automation 3 (3) (1987) 249–265.

[16] S.P. Engelson, D.V. McDermott, Error correction in mobile robot map learning, in: Proc. IEEE International Conference on Robotics and Automation, Nice, France, 1992, pp. 2555–2560.

[17] M. Franz, B. Schölkopf, H.A. Mallot, H.H. Bülthoff, Learning view graphs for robot navigation, Autonomous Robots 5 (1998) 111–125.

[18] A. Gelb, Applied Optimal Estimation, MIT Press, Cambridge, MA, 1974.

[19] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Generation Computing 9 (1991) 365–385.

[20] E. Mark Gold, Complexity of automaton identification from given data, Inform. and Control 37 (1978) 302–320.

[21] S. Gopal, R.L. Klatzky, T.R. Smith, Navigator: A psychologically based model of environmental learning through navigation, J. Environmental Psychology 9 (1989) 309–331.

[22] S. Koenig, R. Simmons, Passive distance learning for robot navigation, in: Proceedings of the Thirteenth International Conference on Machine Learning (ICML), Bari, Italy, 1996, pp. 266–274.

[23] D. Kortenkamp, E. Chown, S. Kaplan, Prototypes, locations, and associative networks (PLAN): Towards a unified theory of cognitive mapping, Cognitive Sci. 19 (1995) 1–51.
[24] B. Kuipers, P. Beeson, Bootstrap learning for place recognition, in: Proc. AAAI-02, Edmonton, AB, AAAI Press, 2002, pp. 174–180.
[25] B. Kuipers, Y.T. Byun, A robust qualitative method for robot spatial learning, in: Proc. AAAI-88, St. Paul, MN, 1988, pp. 774–779.
[26] B. Kuipers, Y.T. Byun, A robot exploration and mapping strategy based on semantic hierarchy of spatial representations, Robotics Autonomous Syst. 8 (1991) 47–63.
[27] B. Kuipers, T. Levitt, Navigation and mapping in large-scale space, AI Magazine 9 (2) (1988) 25–43.
[28] B. Kuipers, R. Froom, W.Y. Lee, D. Pierce, The semantic hierarchy in robot learning, in: J. Connell, S. Mahadevan (Eds.), Robot Learning, Kluwer Academic, Dordrecht, 1993, pp. 141–170.
[29] B. Kuipers, D. Tecuci, B. Stankiewicz, The skeleton in the cognitive map: A computational and empirical exploration, Environment and Behavior 35 (1) (2003) 80–106.
[30] B. Kuipers, Modeling spatial knowledge, Cognitive Sci. 2 (1978) 129–153.
[31] B. Kuipers, The spatial semantic hierarchy, Artificial Intelligence 119 (2000) 191–233.
[32] B.C. Kuo, Automatic Control Systems, Fifth Edition, Prentice-Hall, Englewood Cliffs, NJ, 1987.
[33] W.Y. Lee, Spatial semantic hierarchy for a physical mobile robot, PhD Thesis, The University of Texas at Austin, 1996.
[34] D. Leiser, A. Zilbershatz, THE TRAVELLER: A computational model of spatial network learning, Environment and Behavior 21 (4) (1989) 435–463.
[35] V. Lifschitz, Circumscription, in: Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3, Oxford University Press, Oxford, 1994, pp. 297–352.
[36] V. Lifschitz, Nested abnormality theories, Artificial Intelligence 74 (2) (1995) 351–365.
[37] K. Lynch, The Image of the City, MIT Press, Cambridge, MA, 1960.
[38] H.A. Mallot, S. Gillner, Route navigating without place recognition: What is recognized in recognition-triggered responses? Perception 29 (2000) 43–55.
[39] J. McCarthy, S. Buvač, Formalizing context (expanded notes), in: A. Aliseda, R.J. van Glabbeek, C. Westerstähl (Eds.), Computing Natural Language, in: CSLI Lecture Notes, Vol. 8L, 1998, pp. 13–50.
[40] D.V. McDermott, E. Davis, Planning routes through uncertain territory, Artificial Intelligence 22 (1984) 107–156.
[41] P. Moutarlier, R. Chatila, Stochastic multisensory data fusion for mobile robot location and environment modelling, in: Proc. 5th International Symposium on Robotics Research, 1989, pp. 85–89.
[42] A. Musto, K. Stein, K. Schill, A. Eisenkolb, W. Brauer, Qualitative motion representation in egocentric and allocentric frames of reference, in: Proc. Fourth International Conference on Spatial Information Theory (COSIT'99), Springer, Berlin, 1999.
[43] I. Niemelä, P. Simons, Smodels—An implementation of the stable model and well-founded semantics for normal logic programs, in: Proc. 4th International Conference on Logic Programming and Nonmonotonic Reasoning, in: Lecture Notes in Computer Science, Vol. 1265, Springer, Berlin, 1997, pp. 420–429.
[44] M. O'Neill, A biologically based model of spatial cognition and wayfinding, J. Environmental Psychology 11 (1991) 299–320.
[45] A.L. Peressini, F.E. Sullivan, K.J. Uhl, The Mathematics of Nonlinear Programming, Springer, New York, 1988.
[46] J. Piaget, B. Inhelder, The Child's Conception of Space, Norton, New York, 1967, First published in French, 1948.
[47] E. Remolina, B. Kuipers, Towards a formalization of the Spatial Semantic Hierarchy, in: Proc. Fourth Symposium on Logical Formalizations of Commonsense Reasoning, London, 1998.
[48] E. Remolina, B. Kuipers, A logical account of causal and topological maps, in: Proc. IJCAI-01, Seattle, WA, AAAI Press, Menlo Park, CA, 2001, pp. 5–11.
[49] E. Remolina, A logical account of causal and topological maps, PhD Thesis, The University of Texas at Austin, 2001.
[50] R.L. Rivest, R.E. Schapire, A new approach to unsupervised learning in deterministic environments, in: Proceedings of the Fourth International Workshop on Machine Learning, 1987.
[51] B. Schölkopf, H. Mallot, View-based cognitive mapping and path planning, Adaptive Behavior 3 (1995) 311–348.

[52] M.P. Shanahan, Noise and the common sense informatic situation for a mobile robot, in: Proc. AAAI-96, Portland, OR, 1996, pp. 1098–1103.
[53] H. Shatkay, L. Kaelbling, Learning topological maps with weak local odometry information, in: Proc. IJCAI-97, Nagoya, Japan, 1997.
[54] A.W. Siegel, S. White, The development of spatial representations of large-scale environments, in: H. Reese (Ed.), Advances in Child Development and Behavior, Vol. 10, Academic Press, New York, 1975, pp. 9–55.
[55] R. Simmons, S. Koenig, Probabilistic robot navigation in partially observable environments, in: Proc. IJCAI-95, Montreal, Quebec, 1995.
[56] R. Smith, P. Cheeseman, On the representation of and estimation of spatial uncertainty, Internat. J. Robotics Res. 5 (1986) 56–68.
[57] S.D. Steck, H.A. Mallot, The role of global and local landmarks in virtual environment navigation, Presence 9 (1) (2000) 69–83.
[58] J. Tardos, J. Neira, P. Newman, J. Leonard, Robust mapping and localization in indoor environments using sonar data, Internat. J. Robotics Res. 21 (6) (2002) 311–330.
[59] S. Thrun, S. Gutmann, D. Fox, W. Burgard, B. Kuipers, Integrating topological and metric maps for mobile robot navigation: A statistical approach, in: Proc. AAAI-98, Madison, WI, 1998, pp. 989–995.
[60] S. Thrun, Learning metric-topological maps for indoor mobile robot navigation, Artificial Intelligence 99 (1) (1998) 21–71.