Reasoning about Space in SNePS

Pedro Moniz
mpc@ursula.inesc.pt

Miguel Correia
mpc@ursula.inesc.pt

Nuno J. Mamede
njm@inesc.pt

Instituto Superior Técnico / INESC
Technical University of Lisbon
Apartado 13069
1000 Lisboa, Portugal

Abstract

The development of non-trivial domain knowledge representation and reasoning, as naive physics, is becoming an important task of AI. One of the challenges that emerge in this domain is the simulation of physical systems, which requires a lot of theorem proving, planning, constraint satisfaction and consistency checking. We have used a general purpose semantic network system to implement a spatial logic based on connectivity and outlined a basic support for qualitative simulation.

1 Introduction

It has been pointed out by several authors [Randell and Cohn 89] that much of the discourse about the every day world exploits topological and metrical rather than geometrical information. Expressions like "side by side", "just after" or "between" are more common then "the car with its right side parallel to the mail box" or "the round box". The topological information is important for the description of the surrounding world in natural language. So it is natural to think that a formalism that can capture this kind of information and support deduction should be very important to AI. Randell, Cohn and Cui developed a logic formalism, based on the dyadic relation "connectivity", which can be used to reason about space [Randell, Cui and Cohn 92].

Our work is an implementation of this spatial logic [Randell and Cohn 89] [Randell, Cui and Cohn 92] [Cui, Cohn and Randell 92] in SNePS (Semantic Network Processing System) - allowing both representation and reasoning about space. It is also a framework for the implementation of a qualitative simulation system that could infer the transactions of a physical system between an initial and a final state.

SNePS [Shapiro 79] [Shapiro and Rapaport 87] [Shapiro 91] [Shapiro and Rapaport 91] is a knowledge representation and reasoning system based on the semantic net paradigm. Its emphasis is on chaining with rules in "common-sense" reasoning situations as well as a theorem prover. The SNePS utilization in a practical case as the present one is extremely important to reveal problems and suggest new directions of development.

The structure of this paper is as follows. Section 2 describes the space ontology: the set of logical relations that allow description and reasoning about space entities; the relational lattice that defines their subsumption hierarchy; an underlying theory for qualitative simulation; and a transitive table between logical relations. Section 3 describes how the space ontology previously described was implemented in SNePS. First we present one possible representation of the space ontology in SNePSLOG and point out some restrictions imposed by SNIP (the SNePS Inference Package) and ways to overcome them. Then we show our final representation, conditioned by the present technological constraints, which can be used

to perform topological deductions, to prove topological theorems or to make spatial simulations. Section 4 contains some examples.

2 The formalism of the Space Ontology

2.1 The Spatial Logic

A spatial logic is a set of logical relations that allow description and reasoning about space entities. The space ontology implemented by us in SNePS is a small subset of the theory described by Randell, Cui and Cohn in [Randell, Cui and Cohn 92]. This subset is described by Randell and Cohn in [Randell and Cohn 89] and it is based on former work by Clarke.

The space ontology is based on a logical description of the space using regions and their connections. This is a simplification of the theory presented by Randell, Cui and Cohn that, beside regions, also includes physical objects and other types of entities. Until now, only the logic of regions has been sufficiently described [Randell and Cohn 89] [Randell, Cui and Cohn 92]. Although regions can have both a spatial and a temporal interpretation, in our work they are always interpreted as spatial regions.

When two regions share a common point, it is said they are connected, represented by the primitive relation - $C(x, y)$ - read as "x connects with y", which is reflexive and symmetric.

From the basic connection relation, a set of dyadic relations is defined to describe different kinds and specialization of connections. These relations are: $DC(x, y)$ (read as "x is disconnected from y"); $P(x, y)$ ("x is part of y"); $PP(x, y)$ ("x is proper part of y"); $EQUAL(x, y)$ ("x is identical with y"); $PO(x, y)$ ("x partially overlaps y"); $EC(x, y)$ ("x is externally connected with y"); $TPP(x, y)$ ("x is tangential proper part of y") and $NTPP(x, y)$ ("x is non tangential proper part of y"). These relations can be defined from the basic connection relation by the following set of axioms:

$DC(x, y) \equiv \neg C(x, y)$
$P(x, y) \equiv \forall z[C(z, x) \rightarrow C(z, y)]$
$PP(x, y) \equiv P(x, y) \wedge \neg P(y, x)$
$EQUAL(x, y) \equiv P(x, y) \wedge P(y, x)$
$O(x, y) \equiv \exists z[P(z, x) \wedge P(z, y)]$
$PO(x, y) \equiv O(x, y) \wedge \neg P(x, y) \wedge \neg P(y, x)$
$DR(x, y) \equiv \neg O(x, y)$
$TPP(x, y) \equiv PP(x, y) \wedge \exists z[EC(z, x) \wedge EC(z, y)]$
$NTPP(x, y) \equiv PP(x, y) \wedge \neg \exists z[EC(z, x) \wedge EC(z, y)]$
$EC(x, y) \equiv C(x, y) \wedge \neg O(x, y)$

2.2 The Hierarchy

The relations introduced in the former section, can be embedded in a relational lattice that define theirs subsumption hierarchy (Figure 1). For example, if we assume that the region a is a proper part of region b ($PP(a, b)$) then a is either a tangential or a non tangential proper part of b ($TPP(a, b)$ or $NTPP(a, b)$). Likewise, if $TPP(a, b)$ then $PP(a, b)$, and if $EQUAL(a, b)$ then both relations $P(a, b)$ and $P(b, a)$ hold.

2.3 Simulations

The Spatial Logic has been used as the underlying theory in qualitative simulation [Cui, Cohn and Randell 92] where a state is defined by a set of atomic formulas. The simulation takes an initial state that evolves according to envisioning axioms describing direct topological transitions that can be made between pairs of regions. Such evolving is done until a final state is reached. A representation of the topological transitions can be found in Figure 2.

In this context, the basic theory also includes the topological transitions of the base relations,

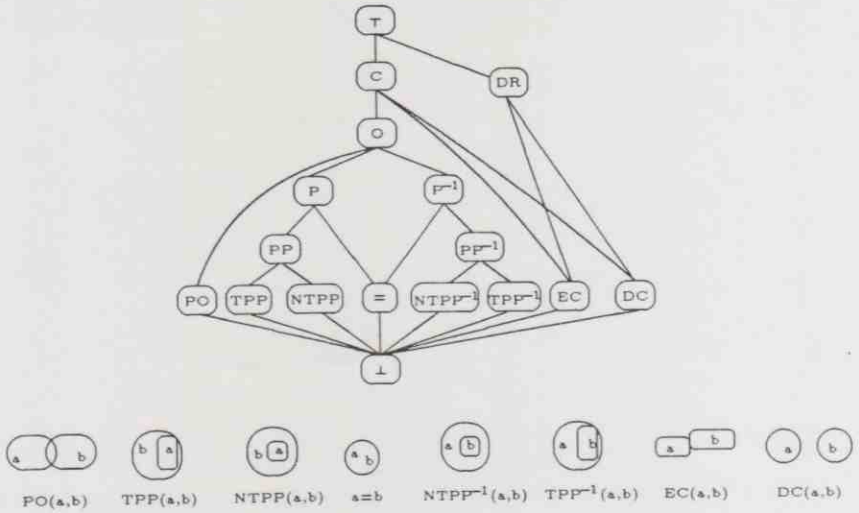

Figure 1: A lattice defining a subsumption hierarchy of the dyadic relations defined solely in terms of the primitive relation $C(x, y)$.

constraints that apply within a state and between adjacent states, add-rules that sanction the introduction of new objects into the domain of the next state, and delete-rules to ratify the elimination of objects in the next state.

To make a simulation possible it is important that the consistency of the initial state description be checked along the envisioning process. This is done using a transitivity table. In the next section transitivity tables will be discussed. According with [Randell and Cohn 89] the base relations and their topological transitions are depicted in Figure 2.

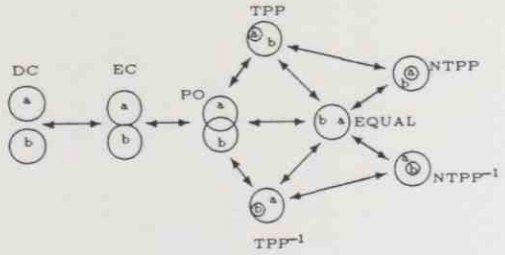

Figure 2: Base relations and their topological transitions.

2.4 Transitivity Tables

One of the problems that will be referred to later is the need of consistency checking of a set of formulas. One way to do this, is to prove the transitivity between pairs of formulas. Suppose that we have a set of relations $P = \{P_{y1}(x_1, x_2), ..., P_{yn}(x_i, y_j)\}$ between regions ($P(a, b)$ reads "region a has a P connection with region b") describing the state of a system modeled by regions and their connective relations. For each pair of relations that hold in P, for example $P_1(a, b)$ and $P_2(b, c)$, we can ask which relations $P_j(a, c)$ or $P_i(c, a)$ we can have in P and still have a logically consistent set P. This is a typical problem in theorem proving.

There are several strategies to construct such tables, but they are all complex and tedious and in some cases even too difficult to secure. These difficulties, along with the necessity of using transitivity tables, give a major incentive to the development of a theorem prover that can generate the entries for

$R_1(a,b)$ \ $R_2(b,c)$	DC	EC	PO	TPP	$NTPP$
DC	any	DR, PO, PP	DR, PO, PP	DR, PO, PP	DR, PO, PP
EC	DR, PO, PP^{-1}	DR, PO, TPP, TPP^{-1}	DR, PO, PP	EC, PO, PP	PO, PP
PO	DR, PO, PP^{-1}	DR, PO, P^{-1}	any	PO, PP	PO, PP
TPP	DC	DR	DR, PO, PP	$NTPP$	PP
$NTPP$	DC	DC	DR, PO, PP	$NTPP$	$NTPP$

Table 1: Transitivity Table (from [Randell, Cohn and Cui 92]).

such tables. Such a theorem prover would be a significant progress in automated theorem proving. Our work in SNePS is an attempt to solve this problem using the capability of representation and reasoning of this system. Some examples shown in Section 4 are entries of this table.

3 SNePS Implementation

3.1 SNePS Restrictions and Implementations Issues

Using the expressiveness of SNePSLOG [Matos and Martins 89] [Shapiro *et al.* 93] it is possible to translate the space formalism expressions almost directly.

```
all(x)(region(x) => C(x,x))
all(x,y)({region(x),region(y)} &=> {C(x,y) <=> C(y,x)})
all(x,y)({region(x),region(y)} &=> {DC(x,y) <=> DC(y,x)})
all(x,y)({region(x),region(y)} &=> {PP(x,y) <=> ~PP(y,x)})
all(x,y)({region(x),region(y)} &=> {O(x,y) <=> O(y,x)})
all(x,y)({region(x),region(y)} &=> {DR(x,y) <=> DR(y,x)})
all(x,y)({region(x),region(y)} &=> {PO(x,y) <=> PO(y,x)})
all(x,y)({region(x),region(y)} &=> {EC(x,y) <=> EC(y,x)})
all(x,y)({region(x),region(y)} &=> {TPP(x,y) <=> ~TPP(y,x)})
all(x,y)({region(x),region(y)} &=> {NTPP(x,y) <=> ~NTPP(y,x)})
all(x,y)({region(x),region(y)} &=> {DC(x,y) <=> (~C(x,y))})
all(x,y)({region(x),region(y)} &=> {P(x,y) <=> (all(z) ({region(z),C(z,x)} &=> {C(z,y)}))})
all(x,y)({region(x),region(y)} &=> {PP(x,y) <=> (P(x,y) and ~P(y,x))})
all(x,y)({region(x),region(y)} &=> {EQUAL(x,y) <=> (P(x,y) and P(y,x))})
all(x,y)({region(x),region(y)} &=> {O(x,y) <=> exists(z)(region(z) and P(z,x) and P(z,y))})
all(x,y)({region(x),region(y)} &=> {PO(x,y) <=> (O(x,y) and ~P(x,y) and ~P(y,x))})
all(x,y)({region(x),region(y)} &=> {DR(x,y) <=> ~O(x,y)})
all(x,y)({region(x),region(y)} &=> {TPP(x,y) <=> exists(z)(region(z) and PP(x,y) and EC(z,x) and EC(z,y))})
all(x,y)({region(x),region(y)}&=>{NTPP(x,y)<=> exists(z)(region(z)and PP(x,y) and (~EC(z,x)or ~EC(z,y))})})
all(x,y)({region(x),region(y)} &=> {EC(x,y) <=> (C(x,y) and ~O(x,y))})
```

These expressions although correct according to SNePSLOG syntax, do not allow the SNIP (the SNePS Inference Package) to do many of the expected deductions. We are going to analyze why this happens and how to avoid this problem.

Negation

The use of negations can cause problems. For example, if SNePS knows `C(a,b)`, it still cannot answer the query `~C(a,b)?`. To solve this problem we can add the second-order expression:

```
all(r,x,y) (r(x,y) <=> (~(~r(x,y))))
```

or we can substitute a simple expression like:

```
all(x,y)({region(x),region(y)} &=> {C(x,y) <=> ~DC(x,y)})
```

with: `all(x,y)({region(x),region(y)} &=> {C(x,y) <=> ~DC(x,y)})`

```
all(x,y)({region(x),region(y)} &=> {DC(x,y) <=> ~C(x,y)})
```

This solution, although less elegant is more efficient than the previous one which causes much additional inference.

Existential Quantifiers
Since the existential quantifiers are not implemented in SNePS (yet), they have to be substituted by Skolem functions folloing the usual way. For example, the formula:

```
all(x,y)({region(x),region(y)} &=> {TPP(x,y) <=> exists(z)(region(z) and PP(x,y) and EC(z,x) and EC(z,y))})
```

is replaced by:

```
1. all(x,y) ({region(x),region(y)} &=>
            {TPP(x,y) => {PP(x,y) and region(SKF_TPP(x,y)) and EC(SKF_TPP(x,y),x) and EC(SKF_TPP(x,y),y)}})

2. all(x,y,z) ({region(x),region(y),region(z)} &=> {{PP(x,y),EC(z,x),EC(z,y)} &=> {TPP(x,y)}})
```

During the our experiments in theorem proven (see example 3 in section 4) with the spatial logic, in order to perform some deduction we needed a mechanism that could recognize an instance of a Skolem function as a variable (region) already in use. For example, if we know that a region A is a tangential proper part of region B therefore exists an region X that is external connected with B and with A. If we knew that an region C is external connected with region A, it's only natural to think that this region could be the region X, so instead of instantiated the region X as Skolem(A,B) we identified X with the region C. This is a very sensible domain because logic consistency can be violated, so it must be used with care. In our case this kind of reasoning was very useful.

To make this kind of reasoning possible we used the following set of formulas:

```
; unification of the Skolem function with a region...
all(r,f,x,y,z)({skf(r),f(r(x,y),x),f(z,y),region(x),region(y),region(z),region(r(x,y))}&=>{unif(r(x,y),z)})
all(r,f,x,y,z)({skf(r),f(r(x,y),y),f(z,y),region(x),region(y),region(z),region(r(x,y))}&=>{unif(r(x,y),z)})
all(r,f,x,y,z)({skf(r),f(x,r(x,y)),f(z,y),region(x),region(y),region(z),region(r(x,y))}&=>{unif(r(x,y),z)})
all(r,f,x,y,z)({skf(r),f(y,r(x,y)),f(z,y),region(x),region(y),region(z),region(r(x,y))}&=>{unif(r(x,y),z)})

;...the replication of nodes using the unification done previously ...
all(g,x,y,z) ({unif(x,y), g(x,z)} &=> {g(y,z)})
all(g,x,y,z) ({unif(x,y), g(z,x)} &=> {g(z,y)})

;.. and  the following predicates must exist "a priori".
skf(skolem_function_name(a,b))
region(skolem_function_name(a,b))
```

3.2 The Hierarchy of Relations in SNePSLOG

The formulas that can be written by the simple inspection of the hierarchy of relations, depicted in Figure 1, gives birth to a second set of formulas, redundant to the first set. This set of formulas is more efficient in the inference of topological relations although is not as powerful as the first one. The hierarchy relations in SNePS are:[1]

```
;First level of the hierarchy
all(x,y)({region(x),region(y)} &=> {C(x,y) => andor(1,1){O(x,y), EC(x,y)}})
all(x,y)({region(x),region(y)} &=> {O(x,y) => C(x,y)})
all(x,y)({region(x),region(y)} &=> {EC(x,y) => C(x,y)})
all(x,y)({region(x),region(y)} &=> {DR(x,y) => andor(1,1){EC(x,y), DC(x,y)}})
all(x,y)({region(x),region(y)} &=> {EC(x,y) => DR(x,y)})
all(x,y)({region(x),region(y)} &=> {DC(x,y) => DR(x,y)})

;Second level of the hierarchy
all(x,y)({region(x),region(y)} &=> {O(x,y) => andor(1,1){P(x,y),P(y,x),PO(x,y)}})
all(x,y)({region(x),region(y)} &=> {P(x,y) => O(x,y)})
all(x,y)({region(x),region(y)} &=> {P(y,x) => O(x,y)})
all(x,y)({region(x),region(y)} &=> {PO(x,y) => O(x,y)})

;Third level of hierarchy
all(x,y)({region(x),region(y)} &=> {P(x,y) => andor(1,1){PP(x,y),EQUAL(x,y)}})
all(x,y)({region(x),region(y)} &=> {PP(x,y) => P(x,y)})
```

[1] The andor(1,1) connective implements the "exclusive or" in SNePSLOG.

```
all(x,y)({region(x),region(y)} &=> {{EQUAL(x,y)} => andor(1,1){P(x,y),P(y,x)}})

;Fourth level of the hierarchy
all(x,y)({region(x),region(y)} &=> {PP(x,y) => andor(1,1){TPP(x,y), NTPP(x,y)}})
all(x,y)({region(x),region(y)} &=> {TPP(x,y) => PP(x,y)})
all(x,y)({region(x),region(y)} &=> {NTPP(x,y) => PP(x,y)})
```

3.3 Implementation of the Spatial Logic According to SNePS Technological Constraints

This section lists the SNePSLOG formulas of the spatial logic used in topological deductions and theorem proving. Our goal was to prove spatial topology theorems based on these formulas. This goal was partially achieved, SNePS proved the simpler theorems.

The formula list is divided into two groups. The first implements the axioms of the relations. The second contains the rules that allow the deduction of a relation from other relations. These formulas are obtained from two sources: The formal definition of the relation and their subsumption hierarchy.

```
;Connectivity axioms - C
all(x)({region(x)} &=> {C(x,x)})
all(x,y)({region(x),region(y)} &=> {C(x,y) => C(y,x)})

;axioms of derived relations
all(x,y)({region(x),region(y)} &=> {DC(x,y) => DC(y,x)})
all(x,y)({region(x),region(y)} &=> {PP(x,y) => ~PP(y,x)})
all(x,y)({region(x),region(y)} &=> {O(x,y) => O(y,x)})
all(x,y)({region(x),region(y)} &=> {DR(x,y) => DR(y,x)})
all(x,y)({region(x),region(y)} &=> {PO(x,y) => PO(y,x)})
all(x,y)({region(x),region(y)} &=> {EC(x,y) => EC(y,x)})
all(x,y)({region(x),region(y)} &=> {TPP(x,y) => ~TPP(y,x)})
all(x,y)({region(x),region(y)} &=> {NTPP(x,y) => ~NTPP(y,x)})

; Relation definitions
;connected - C  (the primitive relation)
all(x,y)({region(x),region(y)} &=> {C(x,y) => andor(1,1){O(x,y), EC(x,y)}})

;disconnected - DC
all(x,y)({region(x),region(y)} &=> {DC(x,y) => (~C(x,y))})
all(x,y)({region(x),region(y)} &=> {C(x,y) => (~DC(x,y))})
all(x,y)({region(x),region(y)} &=> {(~DC(x,y)) => C(x,y)})
all(x,y)({region(x),region(y)} &=> {(~C(x,y)) => DC(x,y)})
all(x,y)({region(x),region(y)} &=> {DC(x,y) => DR(x,y)})

;part - P
all(x,y,z)({region(x),region(y),region(z)} &=> {P(x,y) => (~C(z,x) or C(z,y))})
all(x,y,z)({region(x),region(y),region(z)} &=> {{P(x,y), C(z,x)} &=> {C(z,y)}})
all(x,y,z)({region(x),region(y),region(z)} &=> {{P(x,y), ~C(z,y)} &=> {~C(z,x)}})
all(x,y)({region(x),region(y)} &=> {P(x,y) => O(x,y)})
all(x,y)({region(x),region(y)} &=> {P(x,y) => andor(1,1){PP(x,y), EQUAL(x,y)}})

;equal - EQUAL
all(x,y)({region(x),region(y)} &=> {{EQUAL(x,y)} &=> {P(x,y),P(y,x)}})
all(x,y)({region(x),region(y)} &=> {{P(x,y), P(y,x)} &=> {EQUAL(x,y)}})

;proper part - PP
all(x,y)({region(x),region(y)} &=> {{PP(x,y)} &=> {P(x,y), ~P(y,x)}})
all(x,y)({region(x),region(y)} &=> {PP(x,y) => andor(1,1){TPP(x,y),NTPP(x,y)}})
all(x,y)({region(x),region(y)} &=> {{P(x,y), ~P(y,x)} &=> {PP(x,y)}})

;overlaps - O
all(x,y)({region(x),region(y)} &=>{{O(x,y)} &=> {region(SKF_O(x,y)),P(SKF_O(x,y),x),P(SKF_O(x,y),y)}})
all(x,y,z)({region(x),region(y),region(z)} &=> {{P(z,x), P(z,y)} &=> {O(x,y)}})
all(x,y)({region(x),region(y)} &=> {O(x,y) => C(x,y)})
all(x,y)({region(x),region(y)} &=> {O(x,y) => andor(1,1){PO(x,y),P(x,y)or P(y,x)}})
```