A Calculus for Reasoning About Containment and Object Access

Michael Pool

Information Extraction and Transport, Inc.
1911 North Fort Myer Dr., Suite 600
Arlington, VA 22209
703-841-3500, 703-841-3501(fax)
mpool@iet.com
Keywords: container reasoning, qualitative spatial reasoning, molecular biology, logic, spatial accessibility

Abstract

In this paper we analyze two different notions of containment that are relevant to reasoning about physical systems, a notion of being inside versus a notion of being restricted. We develop a formal vocabulary that allows us to represent and reason about restrictive containment and formalize three kinds of accessibility that are each salient to attempts to reason about the ability of pairs of objects in a system to interact. We also consider the relation of this calculus to the RCC-8 and potential applications.

Introduction

In this paper we discuss a formal representation of a physical system with respect to the containers and barriers that exist within the system. The formalism allows us to reason efficiently about the accessibility relationships between objects within the system. The formalism is intended to be compatible with existing qualitative spatial reasoning methods. It is motivated by a desire to extend work that has been done to represent central notions in biological structure (Cohn 2001) and biological process simulation (Cui Cohn Randell 1992). However, we anticipate that the formalism is sufficiently general to be applicable to any system in which containment is relevant, e.g., computer network security, building security analysis, etc.

Containment

Before we introduce containment vocabulary, it is useful to consider two notions of containment that are relevant to reasoning about physical systems. The first is a strictly spatial notion concerning the location of one object with respect to the boundaries of another. Something contains something else, depending on the context, if and only if the object is inside, located within the convex hull of, encircled by, wrapped by, or "located within" the second object. In this sense of containment, a car contains passengers, playgrounds contain children and a stew contains potatoes, etc. Let us call this the locational sense of containment. However, there is a second notion of containment corresponding to the idea that a prison contains a prisoner in a way that it does not contain a prison guard. This sense of containment has to do not just with being located inside something else but also with being constrained from exiting the object within which it is inside. Let us refer to this as the restrictive sense of containment. The restrictive sense of containment should be of interest in attempts to model dynamic physical systems. Useful models of such systems may require the ability to represent and reason about the accessibility of objects or spaces within the system. This calculus is intended for such knowledge representation efforts.

In the rest of this paper, unless otherwise noted, we use 'containment' in this specialized "restrictive" sense. When we intend the locational sense of containment we simply use 'inside' or, if absolutely necessary, refer to "locational containment" to disambiguate from the restrictive notion of "contains".

Before launching into the formalization, it is useful to consider the representational needs that motivate this effort. Often containment requires considerations over and above simple considerations of relative proximity as formalized for instance in the RCC-8 (Cohn, Bennett, Gooday, Gotts, 1997) or even quantitative size considerations. Cell biology provides interesting examples such as cases of diffusion across lipid bilayers. In a standard cell biology textbook we encounter claims that small nonpolar molecules, unlike charged molecules, such as molecular oxygen and carbon dioxide, readily diffuse across a lipid bilayer.

> The smaller the molecule and, more importantly, the fewer its favorable interactions with water (that is, the less polar it is), the more rapidly the molecule diffuses across the bilayer. (Alberts 1998)

and

… Lipid bilayers are highly impermeable to all ions and charged molecules, no matter how small. (Alberts 1998)

These examples underscore some important facts about containment. First, whether or not an object contains another object is not solely a function of object size. Hence, it is not possible to reason about containment solely in terms of the size of the objects and the size of the pores in the potential containers. An object's ability to exit another object may be based on its size or features of the contained object that have little to do with size and non-size related features of the container. Note, for instance, that an mRNA molecule in a eucaryotic cell is able to leave the cell after it has undergone capping and polyadenylation. These processes do not decrease the size of the molecule but they alter its stability and ability to pass out of the nucleus intact. A simpler example of how non-spatial changes, or at least changes that wouldn't likely be represented with spatial vocabulary, in the container can change its containment status is unlocking the door to a cell. The point is that state descriptions in a qualitative reasoning system will often have to pay explicit attention to issues of containment and permeability in any given state description. Process descriptions will similarly need to consider how processes affect containment and permeability. It would be difficult to reduce these properties to the properties typically considered in qualitative spatial reasoning.

In the first section below, we present and define the terms that we use for representing the relatively generic spatial relations required to reason about containment and accessibility and we compare and justify this scaled-back spatial representation with the RCC-8. In the second section we discuss the representation of containment and accessibility and how to reason about these notions within a system. We conclude with some comments and suggestions for the implementation of these formalisms within a simulation process or plan reasoning techniques.

We assume a second order logic which, for purposes of facilitating reasoning could be divided into a many-sorted first order universe containing physical objects partitioned into non-containers and potential containers, objects that could contain other objects. However, we do not separate out these sorts here, quantifying over all objects in our domain. The logic required for this representation becomes second-order because we quantify over first-order relations (relations that are defined as sets of n-tuples of first order objects) in two supplementary definitions that we introduce for type-level reasoning.

In our specification below we reserve lowercase variables, from the end of the alphabet, *v,w,x,y,z*, to range over the physical objects and uppercase letters from the middle of the alphabet, *P,Q,R,* as variables ranging over the first-order relations. We use *a,b,c,d* as constants denoting specific physical objects. We denote numbered axioms with an A, (A2) is the second axiom, definitions with a D and provable propositions with a P.

Basic Spatial Relations

A prerequisite for restrictive containment is, of course, that the contained object be in the appropriate spatial relationship with potential container, i.e., that the contained object be inside the potential container. The *inside* relation denotes the relationship of locational containment and as such is quite general by design. Just as we allude to different senses of locational containment when we say that a "building contains people" as compared to when we say that the "salad contains carrots" or especially when we say things like "the firewall contains the entire network", we want to avoid imposing a very particular spatial implementation on the notion of *inside* as we develop it below.

The *inside* relation is given as a primitive of the system. We do not use mereological notions to clarify its intended meaning because we make no assumptions about the kind of simple "insideness" situations that it will be needed for and because it supports our reasoning about containment$_2$ without requiring such support. *inside* is an irreflexive (A1) asymmetric (A2) and transitive (A3) relation as specified in 1-3 below:

(A1) $\forall x[\neg inside(x,x)]$

(A2) $\forall x \forall y[inside(x,y) \rightarrow \neg inside(y,x)]$

(A3) $\forall x \forall y \forall z[[inside(x,y) \wedge inside(y,z)] \rightarrow inside(x,z)]$

The second relevant relation is the 'outside' relation. Intuitively, it is the opposite of being inside and we detail axioms constraining its use below. A good comparison point here is the OUTSIDE relation presented in (Randell et al 1992). In order to keep the vocabulary generally applicable we leave our definition more general but it should not be inconsistent with the more specific definition. We discuss this further below. We intend that *outside* apply to most situations in which two neither of two distinct objects bear the inside relation to the other, but see *overlaps* below. The outside relation is irreflexive, above, and symmetric. Also, *outside(x,y)* is inconsistent with *inside(x,y)*, *i.e.*, *inside* and *outside* cannot simultaneously hold of the same pair of objects.

(A4) $\forall x[\neg outside(x,x)]$

(A5) $\forall x \forall y[outside(x,y) \rightarrow outside(y,x)]$

(A6) $\forall x \forall y[inside(x,y) \rightarrow \neg outside(x,y)]$

For completeness sake, we introduce three other basic spatial relations as well: *overlaps*, *equals*, and $inside^{-1}$.

overlaps is irreflexive (A6), symmetric (A7) and neither transitive nor antitransitive. Also,

(A7) □x[□overlaps(x,x)]

(A8) □x□y [overlaps(x,y) □ overlaps(y,x)]

(A9) □x□y[overlaps(x,y) □ [□inside(y,x) □ □outside(y,x)]]

Typically, a necessary condition for the *overlaps* relation holding of two distinct objects is that some proper part of one of the object be found within the region defined by the convex hull of the other. *equals* is the equality relationship and is, of course, reflexive, symmetric and transitive.

(A10) □x[equals(x,x)]

(A11) □x□y[equals(x,y) □ equals(y,x)]

(A12) □x□y□z[[equals(x,y) □ equals(y,z)] □ equals(x,z)]

(A13) □x□y[equals(x,y) □ [□overlaps(x,y) □ □inside(x,y) □ □outside(x,y)]]

We introduce the relationship of inverse insideness, *inside*$^{-1}$, so as to round out the set of basic relations. It can be defined, however, in terms of the inside relation.

(D1) inside^{-1}*(x,y)* $\equiv_{def}$ *inside(y,x)*

(A14) □x□y[inside^{-1}*(x,y) □ [□overlaps(x,y) □ □equals(x,y) □ □outside(x,y)]]*

(D2) atLeastPartiallyInside(x,y) ≡def [inside(x,y) overlaps(x,y)]

From (D1) and (A1-A3) it follows that inside^{-1} is irreflexive, asymmetric and transitive. The result is a set of relations, {*inside, inside*$^{-1}$*, outside, equals, overlaps*}, that is pair-wise disjoint and jointly exhaustive, *i.e*.,

(A15) □x□y [(inside(x,y) outside(x,y) overlaps(x,y) equals(x,y) inside^{-1}*(x,y))]*

To facilitate reasoning about restrictive containment we introduce three other spatial relations. *dirInside denotes* the relation of being directly inside. *dirOutside* denotes the relation of being directly outside some other object and not encompassed by any other object. The *between* relation is ternary and holds when for a pair of objects there is some third object such that one element of the pair is outside of it while the other is inside it. The precise definition of these relations is below:

(D3) dirInside(x,y) $\equiv_{def}$ *[inside(x,y) □ □□z[atLeastPartiallyInside(z,y) □ inside(x,z)]]*

(D4) between(x,y,z) $\equiv_{def}$ *[inside(x,z) □ outside(y,z)]*

(D5) dirOutside(x,y) $\equiv_{def}$ *[outside(x,y) □ □□z[between(x,y,z)] □ □□v[(between (y,x,v)]]*

Because *inside(x,y)* is implied by *dirInside(x,y)* we can demonstrate its irreflexivity (P1) and asymmetry (P2).[1]

(P1) □x[□dirInside(x,x)]

(P2) □x□y[dirInside(x,y) □ □dirInside(y,x)]

We can demonstrate that it is anittransitive as follows:

Suppose that *dirInside* was not anittransitive, i.e., for some *a,b,c, dirInside(a,b)*, *dirInside(b,c)* and *dirInside(a,c)*. Then *inside(a,c)* by (D3) and □ □*v[inside(v,c)* and *inside(a,v)]*. But, also from (D3) and the given information, *inside(b,c)* and *inside(a,b)*. Hence, there can be no such *a,b,c*. *dirInside* is antitransitive.

(P3) □x□y□z [[dirInside(x,y) □ dirInside(y,z)] □ □dirInside(x,z)]

Given (D5), (A4) and (A5) it is also easy to show that *dirOutside* is irreflexive and symmetric.

Figure 1 illustrates the difference between *dirInside* and *inside*. In the pair of ovals on the left*, dirInside(a,b)* but on the right, □*dirInside(a,b)* because *between(a,b,c)*. Figure 2 helps to clarify the *between* relation. In Scenario 1 of Figure 2, □*between(a,c,b)*, despite its consistency with natural language use of 'between', but the relation would apply to the objects in Scenario 2 because *inside(a,b)* and *outside(c,b)*.

---

[1] For instance, consider the proof for irreflexivity. Suppose that *dirInside* was not irreflexive, i.e., for some *a*, *dirInside(a,a)*. By (D2) *inside(a,a)* but by (A1), □*inside(a,a*. Hence, we can reject the hypothesis that *dirInside* is irreflexive. The proof for the asymmetry of *dirInside* would proceed similarly.

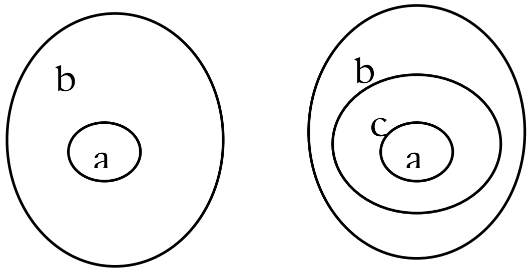

Figure 1: dirInside(a,b) and not dirInside(a,b)

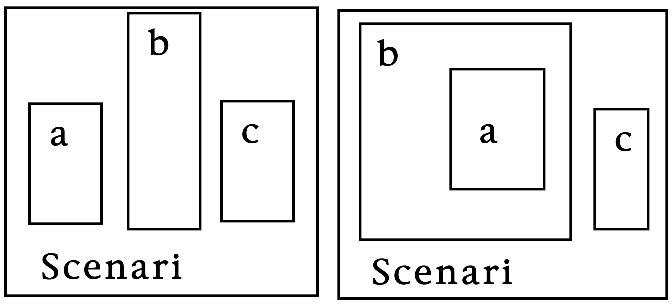

Figure 2: The 'between' relation

Comparison with RCC-8

Readers interested in implementing a more carefully defined notion of the locational sense of 'inside' may want to refer to the notions of insideness as spelled out in (Randell, Cohn and Cui 1992). They offer an axiomatization that captures the typical meaning of 'inside'. Let *RegionFn(x)* denote the region occupied by x in our state description and *ConvHullFn(x)* denote the region encompassed by the convex hull of the object x. Their variables in the original definition range over spatial regions rather than objects so we translate their definition in (R).

(R) *inside(x,y)* $\equiv_{def}$ *[discrete(RegionFn(x),RegionFn(y))* □ *part(RegionFn(x), ConvHullFn(y)]*

Presumably R will be consistent with many applications of the inside relation we offer here but it can be easily abandoned insofar as nothing in the remainder of the accessibility and containment formalization we develop depends on this. In other reasoning contexts, being P-INSIDE (Randell, Chui and Cohn, 1992) may be practically equivalent to inside. Nevertheless, for the purposes of illustrating some of these relations we assume that something like the R definition applies.

For most applications of this calculus, an object *a* is inside an object *b* when all or almost all of *a*'s parts are found inside the boundaries of object *b*. Of course, what we mean by 'almost all', 'part' and 'boundary' is purposefully left unspecified here. It suffices to note that whether or not an object *a* is inside an object *b* in a system at a given point in time will supervene on facts about *a*'s spatial location relative to b at that point in time. The necessary and sufficient conditions for insideness will depend on the kind of accessibility that we're interested in. In fact, perhaps the notion of spatial insideness isn't necessary for accessibility reasoning at all. One could imagine applying the reasoning system below to some network security system for bank accounts in which the notion of insideness is a metaphor for being guarded by some security system (*e.g*., inside the firewall) in which case the spatial notion of insideness is completely irrelevant. Nevertheless, in systems in which we want to reason about containment some kind of insideness will be a prerequisite.

Finally, if we wish to relate this system to a system for reasoning in two dimensions and implement only the RCC-8 relations we might naturally map our relations as follows.

Let us assume that we are considering a fixed system, i.e., a set of objects located at a specific space-time point.

inside(x,y) $\equiv_{def}$ *properPart(RegionFn(x),ConvHullFn(y))*

outside(x,y) $\equiv_{def}$ *discrete(RegionFn(x),RegionFn(y))*

overlaps(x,y) $\equiv_{def}$ *partiallyOverlaps(ConvexHullFn(x),ConvexHullFn(y))*

Note that if we were to implement these above definitions within our calculus we would be able to prove the following assertion. However, we will posit it as an axiom:

(A16) □*x*□*y*□*z[[inside(x,y)* □ *inside(x,z)]*□ □*outside(y,z)]*

In summary, we have purposely kept our suite of spatial representation relations vague so as to facilitate application to a wider variety of applications and because our notions of accessibility do not rely on the ability to make fine grained spatial distinctions about juxtaposition and contact. We also note that the mereological considerations are fairly limited at present and we have done little to consider how mereological consideration affects permeability and accessibility. Finally, we stress that our arguments for our spatial relations are the objects themselves rather than the regions occupied by the objects. This is essential because the permeability of barriers and containers requires consideration of the objects rather than their regions.

Accessibility

Here we turn to the question of accessibility. Informally, when we claim that *b* is accessible to *a* we mean that either there are no objects between, in the sense specified in (D4), *a* and *b* or the objects between *a* and *b* are unable to restrict *a* from getting at *b*. Formalizing the restrictiveness of objects requires the introduction of two more primitive relations, *inPerm* and *outPerm*. *inPerm(x,z)* is intended to represent the fact that the perimeter of z is permeable to x such that x could pass from being directly outside (*directlyOutside*) z to being directly inside (*directlyInside*) z. Similarly, *outPerm(x,z)* is intended to represent the fact that the perimeter of z is permeable to x such that x could pass from being directly intside (*directlyInside*) z to being directly outside (*directlyOutside*) z. Both relations are irreflexive.[2]

(A17) *inPerm(x,y)* $\equiv_{def}$ *□x□y[□inPerm(x,x)]*

(A18) *outPerm(x,y)* $\equiv_{def}$ *□x□y[□outPerm(x,x)]*

We require these extra notions because accessibility concerns not just the spatial configurations but the extent to which the configured objects present a barrier. The materials in the cupboard under the sink become accessible to the toddler if she is able to open the cupboard door, whether the door is closed is irrelevant to accessibility given the door-opening skill.[3] We define permeable(x,y) as follows:

(D6) *permeable(x,z)* $\equiv_{def}$ *[inPerm(x,z) □ outPerm(x,z)]*

In many applications it is not felicitous to insist that either *inPerm(a,b)* or *□inPerm(a,b)* for any two objects *a* and *b* in the system. As our cell biology quotes above indicate, the ability of molecules to diffuse through a cell membrane is a state of affairs that requires representation of degree of permeability. In general, permeability does in fact admit of degree and within systems for which partial permeability is relevant to a state description we may need to implement this system in a Bayesian network so as to allow for the specification of probabilities of penetration for given objects or object types. Our thinking here is that permeability is best understood in terms of breaching. If an object is able to transverse another object only if a breach of some sort has occurred, then we would not deem the object permeable. If, as a default rule, object *a* is able to pass through object *b*, then we might apply the permeability relation. In general we would acknowledge that it is very difficult to reason accurately about containment, especially with respect to fluids and the like, without invoking some uncertainty reasoning but we limit our considerations to logical rules about containment for the present.

We also note the significance of the ability to tease out the inward-outward direction of the permeability relation. It is an important characteristic of many physical systems, not to mention an untold number of television programs involving characters stuck in storage closets, that objects are able to easily pass into a container but are able to leave the container only with great difficulty. In other words, *inPerm(x,z)* and *□outPerm(x,z)*.

Given this notion of access and permeability, we can now define containment as follows:

(D7) *containedBy(x,y)* $\equiv_{def}$ *[inside(x,y) □ □outPerm(x,y)]*

(D8) *blockedBy (x,y)* $\equiv_{def}$ *[outside(x,y) □ □inPerm(x,y)]*

In the sections below, when discussing an object, *y*, we refer to the objects such that *blockedBy(x,y)* as a 'barrier' with respect to y and the objects such that *containedBy(x,y)* as a 'container' with respect to *y*.

These new relations give us a means by which to define accessibility for any two objects. Consider the following relation:

(D9) *inBarrierBetween(x,y,z)* $\equiv_{def}$ *[blockedBy(x,z) □ inside(y,z])*

Assertions of the form *inBarrierBetween(a,b,c)* can be interpreted as "*a* cannot access *b* because *c* is between them and blocks its progress."

(D10) *outContainerBetween(x,y,z)* $\equiv_{def}$ *[containedBy(y,z) □ outside(x,z)]*

Assertions of the form *outContainerBet(a,b,c)* can be interpreted to mean that *b* cannot access *a* because *b* is contained by *c*. Finally we define the notion of accessibility of one object to another:

(D11) *acc(y,x)* $\equiv_{def}$ *□□z[inBarrierBetween(x,y,z) outContainerBetween(y,x,z)]*

As we would expect, from these definitions, we can prove assertions such as the following[4]:

---

[2] If we divide our domain of first order objects into potential containers and non-containers then the second arguments for *inPerm* and *outPerm* should be constrained to be containers.

[3] We might spell out the meaning of these relations in modal terms, e.g., *inPerm(x,y)* means that given the current state of the system, *dirOutside(x,y)* and some z, such that dirInside(z,y), *□[dirInside(z,y) □contact(x,y)]*. However, we leave it as a primitive notion for now.

[4] Proof that □x□y□z[[inPerm(x,z) □ dirInside(y.z) □ dirOutside(x,z)] □ acc(y,x)]
Consider a,b,c, such that inPerm(a,c), dirInside(b,c), dirOutside(a,c). Can we show that acc(b,a)?

(P4) ∀x∀y∀z[[inPerm(x,z) ∧ dirInside(y.z) ∧ dirOutside(x,z)] → acc(y,x)]

(P5) ∀x∀y∀z[[outPerm(x,z) ∧ dirOutside(y.z) ∧ dirInside(x,z)] → acc(y,x)]

The utility of the relations described in (D7)-(D11) is that they allow us to describe the state of a physical system at a given moment in time and analyze it for potential problems and possibilities in terms of the accessibility of certain objects. For example, we can consider vulnerabilities in a computer network, the effectiveness of a security alarm system or perform more accurate *in silico* experiments about biological systems. Given a representation of a system in terms of the inside and outside relation as well as the expression of the capabilities of the objects in the system to easily transverse containers, we are now able to

---

Let us assume that ¬acc(b,a). Hence, let us suppose that there is some d such that [inBarrierBetween(a,b,d) outContainerBetween(b,a,d)] (1)

Suppose that inBarrierBetween(a,b,d)
Then, by (D9) blockedBy(a,d) (2) and inside(b,d) (3). By (D7) and (2), outside(a,d) (4) and ¬inPerm(a,d) (5).
Consider the relationship between d and c. By (A15), either inside(d,c), outside(d,c), inside-1(d,c), equals(d,c), overlaps(d,c).
If equals(d,c), then inPerm(a,d) and ¬inPerm(a,d). So ¬equals(d,c).
From (D2), (3) and the fact dirInside(b,c) we can infer that ¬atLeastPartiallyInside(d,c), i.e., ¬inside(d,c) and ¬overlaps(d,c).
But $inside^{-1}(d,c)$, means that inside(c,d) (D1), so, given (4), we can show that between(c,a,d) contrary to the fact that dirOutside(a,c) which, by (D5) implies that ¬∃z between(c,a,z). So ¬inside-1(d,c).
So, ¬inBarrierBetween(a,b,d).

If (1) and ¬inBarrierBetween(a,b,d), then, outContainerBetween(b,a,d).
Suppose outContainerBetween(b,a,d), then, from (D10), containedBy (a,d) (6) and outside(b,d) (7)
and from (6) and (D7), inside(a,d) (8) and ¬outPerm(a,d) (9).

Consider the relation between d and c.
If ¬equals(d,c), then, from (D4), (7) and (8), between(a,b,d). But dirOutside(a,c), (hyp.)
so, from (D5), ¬∃v[between(a,b,v)]. Hence ¬¬(equals(d,c).

But, suppose equals(d,c). From our initial hypothesis, dirOutside(a,c) and so, from (D5), outside(a,c). But assuming equals(d,c) and (8), we can show inside(a,c). But given (A6), this generates a contradiction. Hence, ¬outContainerBetween(b,a,d).

Hence, the assumption that ¬acc(b,a) leads to a contradiction. So acc(b,a) and we demonstrate the theorem by universal generalization.

pose queries about unidirectional accessibility. For example, if we are concerned about preventing object a from accessing object b, we can ask:

inbarrierBetween(a,b,?x)

In other words, "Which barriers block *a* from coming in to meet *b*?" If we want to query as to whether there is anything that keeps *b* in and prevents it from accessing *a*, we can ask:

outContainerBetween(b,a,?y)

If there are no bindings for either of these we conclude *acc(a,b)*. Note that acc is not a symmetric relationship as there may be objects, z such that *inside(a,z)*, *inPerm(b,z)* but *¬outPerm(a,z)*. In other words, a is accessible to b but b is not accessible to a.

Observe that the above rules concerning accessibility assume that in the attempt to generate a path through containers and barriers, the barriers and containers cannot themselves be used to transport objects in the system and that the accessed object will not itself move so as to become more accessible. However, let us briefly consider how to reason about accessibility in the situations in which:

a) Both objects move: the containers and barriers remain fixed but the two objects of interest could both exploit permeability properties in generating a path to a state in which both objects are accessible to each other. We call this "weak accessibility."

b) Containers move: Some or all of the containers (in the locational sense, at least) and barriers do not remain stationary but are also able to exploit permeability properties and transports container contents such that contained objects become accessible to each other. For example, consider a border crossing that must be crossed by car and not on foot. We might say *outPerm(carA,BorderC)* or *outPermeableType(Car,BorderC)*(where 'Car' denotes the property "being a car", see below). The border is *outPerm* for the car but not for the pedestrian. However, since the car is *inPerm* and *outPerm* for the pedestrian, s/he can use this car to cross the border. Of course, this is the accessibility maneuver exploited in the Trojan Horse story of mythology. Let us call this "indirect accessibility."

Weak Accessibility

Suppose that we now want to consider two objects *a* and *b* from the perspective of determining whether or not they are able to exploit existing permeability properties in the various objects that serve as potential barriers and containers so as to be able to move to a state such that □□[*between(a,b,x)*]. When it is possible for *a* and *b* to get from their initial location by traversing permeable boundaries, then we say that *weaklyAcc(a,b)*.

But when else can we claim that the *weaklyAcc* relation holds between two objects? We need to determine whether there are any objects in the system that are accessible to both of the objects.

(D12) *weaklyAcc(x,y)* $\equiv_{def}$ *□z[[directlyInside(x,z) □ acc(x,y)] □ [directlyInside (y,z) □ acc(y,x)] [[directlyOutside(y,z) □ acc(y,x)] □ [directlyOutside(x,z) □ acc(x,y)]]]*

This definition states that if there is some point in the system accessible to both *a* and *b*, *z* in the definition, then the objects are accessible to each other. From the definition of *weaklyAcc* it is straightforward to prove the symmetry of *weaklyAcc* and that *weaklyAcc(x,y)* is implied by *acc(x,y)* and *acc(y,x)*.

(P6) *□x□y[weaklyAcc(x,y) □ weaklyAcc(y,x)]*

(P7) *□x□y[[acc(x,y) acc(y,x)] □ weaklyAcc(x,y)]*

In the section 'Graph Theory Tests for Accessibility' below we discuss how to articulate a query about weak accessibility in graph theoretic terms.

Indirect Accessibility

The other accessibility problem that we want to address is that of the potential ability of objects to move objects that they contain. Consider Figure 3. Suppose that we wanted to know whether *acc(c,d)*. Further suppose that *outPerm(c,b)*, □*inPerm(d,a)* and that *outPerm(b,a)* but □*outPerm(c,a)*. This means that *c* would not be able to exit *b* and then pass through *a*. However, if *b* were to exit *a* while containing *c*, and then *a* were to exit *b*, *a* would be able to access *d*. Let us call such accessibility, *indirectAccessibility*.

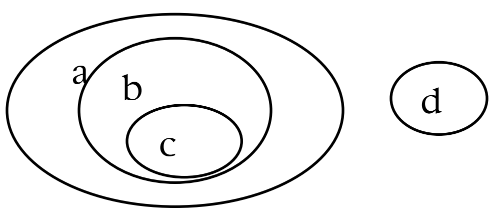

Figure 3: Illustration for explaining Trojan Horse (indirect) accessibility

We define indirect accessibility, *indirectAcc* as follows:

(D13) *indirectAcc(x,y)* $\equiv_{def}$ *□v□z[[[inside(x,v) □ inside(y,z) □ weaklyAcc(y, z)] □ [directlyOutside(x,z) □ acc(y,x)] □ [directlyOutside(y,v) □ acc(x,y)]] [[inside(x,v) □ weaklyAcc(v,y)] □ [directlyOutside(y,v) □ weaklyAcc(x,y)]] [[inside(y,z) □ weaklyAcc(v,x)] □ [directlyOutside(x,z) □ weaklyAcc(x,y)]]]*

The three main disjuncts in this lengthy definition correspond respectively to the situations in which both *x* and *y* could be transported by one of the objects to which they bear the inside relation, the situation in which just x would need to be transported by one of its containers and the situation in which just y would need to be transported by one of its containers. Of course we can imagine this being iterated again so that we determine whether the containers of *x* and *y* are indirectly accessible, etc.

Graph Theory Tests for Accessibility

If our objective is to determine whether or not weak accessibility or indirect accessibility exists we would be hard pressed to use a first order reasoner to efficiently launch a query based on our definitions in (24) and (25). We note that the problem of searching for weak accessibility between two objects can be easily translated into a simple graph theory question. If we're interested in whether or not *a* and *b* are accessible or weakly accessible to each other we take all the objects *z* such that *between(a,b,z)* or *between(b,a,z)* as well as *a* and *b* and make them nodes in a graph. We then insert directed links in the graph in the following way. If *directlyInside(x,y)* and both *inside(b,y)* and *outPerm(b,x)* or both *inside(a,y)* and *outPerm(a,x)* we insert a link from the *x* node to the *y* node. Also, if *directlyOutside(x,y)* and both outside(b,x) and *inPerm(b,y)* or both *outside(a,x)* and *inPerm(a,y)*, we insert a link from *x* to *y*. Given such a representation, whether or not *acc(x,y)* becomes a question of whether there is a path from the *x* node to the *y* node. Whether or not *weaklyAcc(x,y)* is a question of whether or not there is a node z in the graph such that there is a path from *x* to *z* and a path from *y* to *z*. This could be applied to questions about *indirectAccessibility(x,y)* as well by performing the above procedure two or more times, first to the containing$_1$ objects that contain our objects of interest and then to the objects of interest, in accordance with the definition given above.

Relations for Second Order Objects

Of course, much reasoning about accessibility and containment occurs at the type level. For example, that a lipid bilayer can allow a particular water molecule to pass through should be derivable from a more general law relating cells with lipid bilayer membranes and water rather than from an explicit assertion about each water molecule in the system we're representing. Hence, we appeal to second order objects or relations in our domain. We posit relations that hold first between first order objects and unary relations and between pairs of relations.

(D14) *inPermeableType(P,x)* $\equiv_{def}$ □*y[P(y)* □ *inPerm(y,x)]*

and we'd define *outPermeableType(P,x)* in a similar manner, substituting *outPerm* in for *inPerm* in the definition in (26).

This instance-type level definition would be useful if we wanted to note that a particular object has achieved inward or outward permeability for a certain kind of object. For example, a leaky raincoat allows water to go through it. However, permeability is typically easily represented as a relation between types. For example, let the relation C be the property of being a cell, or perhaps being encompassed by a lipid bilayer. Let the relation 'IM' denote the property of being an ionic molecule. Then we note, *inPermeableTypes(IM,C)*. More generally, we define the *outPermeableTypes* in (27) and this applies, mutates mutandis, to *inPermeableTypes* as well.

(D15) *outPermeableTypes(P,Q)* $\equiv_{def}$ □*x*□*y [[P(x)* □ *Q(y)]* □ *outPerm (x,y)]*

Application for Process Reasoning

Finally, we have presented these relations as binary and ternary relations that hold between ordered pairs and ordered triplets of objects. These relations describe system states at a given moment in time. However, it is also important to note that the accessibility and permeability relations are best understood as constraints on how the system can progress over time assuming no changes in permeability relations. For example, when we say that acc(a,b) we mean that given the current state of the system there is a possibility that a future state of the system will have *a* and *b* in contact with each other. Similarly, inPerm(a,b) means that future states requiring a passage from *a* into *b* are not ruled out. As such we want to be more explicit about how these relations might be implemented in a planning or process-reasoning environment. Below we give some examples of how the vocabulary used above might be implemented in actual process reasoning. We indicate how a STRIPS-like planning vocabulary might implement the accessibility vocabulary and some examples of how the vocabulary above might help in representing some example process descriptions in virology and cell biology.

operation: *gainsShell (A, B)* (a new container is generated.)
precondition:
add: *(inside A,B)*

operation: *enters (A,B)* (an object enters another objects)
precondition: (overlaps(A,B) directlyOutside(A,B)), inPerm(A,B)
add: inside(A,B)
delete: outside(A,B)

operation: *losesShell (A,B)* (some container$_1$ or containment$_1$ is removed from the system. This kind of operation is particularly salient to cell biology where membranes often dissolve or dissipate.)
precondition: inside(A,B)
add:
delete: inside(A,B)

operation: *(exits A,B)* (Object A leaves B, but B still exists.)
precondition: *directlyInside(A,B),outPerm(A,B)*
add: *outside(A,B)*
delete: *inside(A,B)*

operation: *losesContainmentStatus (A,B)* (B becomes permeable to A, i.e. will allow A to exit if A is inside B.)
precondition:
add: *outPerm(A,B)*
delete: □*outPerm(A,B)* (if that was stated in the system)

operation: *losesBarrierStatus (A,B)* (B becomes permeable to A, i.e. will allow A to enter if A is outside B.)
precondition:
add: *inPerm(A,B)*
delete: □*inPerm(A,B)* (if that was stated in the system)

More particularly, we also note how aspects of processes in cell biology or virology can be represented by this vocabulary. Virus life cycles require consideration of membrane components and virus types in order to reason about the inPerm relation with respect to the virus and a cell. The viral life cycle involves the formation of several new containers and the destruction or penetration of the old ones and the abilities of various parts of the cell or virus change rapidly with respect to their ability to enter or exit various parts of the cell. Two example passages, Passage A and B, taken from a description of the vaccinia virus life cycle taken from (Flint 2000) are given and represented below:

> **Passage A**: The mechanisms by which vaccinia virus attaches to and enters susceptible host cells are not well understood. The result is release of the core into the cytoplasm, indicating that entry requires fusion of viral with cellular membranes.

To start, we would represent this passage by noting:

inPermeableType (Cell, VacciniaVirus)

where 'Cell' denotes the property of being a cell and 'VacciniaVirus' denotes the property of being a vaccinia virus. The second sentence might be represented as:

□x□y□z [[Cell(x) □ VacciniaVirus(y) □ attached(x,y)] □ inPerm(y,x)]

Passage B: a spherical particle that is believed to possess a double membrane acquired upon wrapping of the membranes of the cellular components of the *cis*-Golgi network about the assembling particle. The virus particle then matures into the brick-shaped intracellular mature virions (IMV), which is released only upon cell lysis.

Two of the salient points from Passage B can be represented as:

□x[SphericalParticle(x) □ □x□y[Membrane(x) □ Membrane(y) □ dirInside(x,y)) □ dirInside(z,y)]

□x□y [[Cell(x) □ inside(y,x) □ IMV(y) □ CellLysis(z) □ patient(x,z)] □ holdsAfter(z, outPerm(y,x)]

Conclusion

Above we have noted that existing approaches to qualitative spatial reasoning allow us to represent and reason about the fact that objects are located inside container objects. However, such representation methods will not allow us to represent containment in the sense of restriction. We have attempted to define this important notion in terms of the objects and permeability properties of physical systems and in terms of different ways in which objects can contain other objects, e.g., inward vs. outward permeability. We have also teased out three different notions of accessibility between objects within a given physical system and suggested methods for investigating whether or not two given objects in a system bear these three kinds of accessibility relations.

We contend that the ability to perform this kind of knowledge representation could be useful in a very wide range of reasoning contexts. This includes computer security, cell biology, virology, building security analysis, chemical storage.

Future work will include an exploration of what we have called "indirect accessibility" with respect to a highly dynamic system in which no objects are stationary. We will also investigate how reasoning about the stationary status of objects and an ontology of physical objects designed in terms of containment capabilities could further facilitate reasoning in this area. As noted above, a further area of investigation concerns the means by which we could integrate notions of partial permeability within such a system.

Acknowledgements

This paper benefited from discussions with and comments from Pierre Grenon, Pierluigi Miraglia and Joshua Powers. Some of this research was supported by a contract (Contract Number F30602-00-C-0173) for DARPA's Rapid Knowledge Formation Project.

References

Alberts B., Bray D., Johnson A., Lewis J., Raff M., Roberts K., Walter P. .1998. *Essential Cell Biology*. New York: Garland.

Cohn A., Bennett B., Gooday J., and Gotts M. (1997). Representing and Reasoning With Qualitative Spatial Relations About Regions. In O. Stock, editor, *Spatial and Temporal Reasoning*, pp. 97-134. Kluwer, Dordrecht.

Cohn, A. 2001. Formalising Bio-spatial Knowledge. In Proceedings of the Second International Conference on Formal Ontology in Information Systems (FOIS'01), C. Welty and B. Smith, eds., 198-209, ACM.

Cui Z., Cohn A.G., Randell D.A. 1992. Qualitative Simulation Based on a Logical Formalism of Space and Time, In Proceedings of the Tenth National Conference on Artifical Intelligence, 679-684. Menlo Park, Calif.: AAAI Press.

Flint, S.J. *et al*. 2000. *Principles of Virology, Molecular Biology, Pathogenesis, and Control.* Washington, D.C.: ASM Press.

Randell D., Cohn A.G., Cui Z. 1992. A Spatial Logic based on Regions and Connection. In Proceedings of the 3rd Int. Conference on Knowledge Representation and Reasoning. 165-176. San Mateo: Morgan Kaufmann.