Integrating a Spatial Reasoner with a Resolution Theorem-Prover*

Thomas R. Ioerger
Department of Computer Science
Texas A&M University
College Station, TX 77843
ioerger@cs.tamu.edu

Abstract

Some spatial reasoning systems use images to solve problems, rather than making formal logical inferences. However, an open question is how to use these systems in contexts where some non-spatial information is also involved. We present a hybrid reasoning method in which we extend the capabilities of a spatial reasoner by integrating it with a resolution theorem-prover. We prove that the hybrid system is refutation-complete, in the sense that, if a domain theory is unsatisfiable, perhaps only because all of its models entail unrealizable images, then our algorithm will halt. We discuss how our approach differs from other hybrid reasoning algorithms in the way it manages the interaction between sub-systems.

Introduction

Recently there has been a resurgence of interest in spatial or diagrammatic reasoning (Glasgow, Narayanan, & Chandrasekaran 1995). Among the various spatial reasoning systems that have been proposed, there are two basic types of approaches: axiomatic and image-based. In axiomatic approaches, the goal is to develop rules that capture the underlying ontology for a class of spatial concepts so that reasoning about knowledge in this form can essentially be accomplished by a theorem-prover (Mukerjee & Joe 1990; Kaufmann 1991). The image-based approaches, however, actually construct some kind of artifact, often an array-based data-structure (Papadias & Glasgow 1991), that represents the described situation, and then makes inferences "by inspection" (Funt 1980).

While image-based systems have been shown to be useful for solving a variety of spatial problems, an open question is how to use them for reasoning in a context where there is also some non-spatial information. For example, suppose an intelligent agent equipped with a spatial reasoning algorithm is faced with the task of reasoning about people at a party. The spatial aspects, such as who is standing next to whom, who is tallest, or what color clothes they are wearing, can be handled by various spatial reasoning algorithms. But it is not clear how to handle non-spatial aspects, such as information about their professions, beliefs, family relationships, etc. Because this information cannot (easily) be depicted, the agent would not be capable of solving an extended class of problems involving mixed spatial and non-spatial reasoning. An example of such a problem would be: "Is the woman in the black dress standing to the left of the Senator at the beverage table married to a Democrat or a Republican?"

One approach to solving these kinds of mixed reasoning problems would be to encode all of the knowledge about the problem in first-order logic, and then use an inference procedure like resolution to answer queries by constructing refutation proofs. However, we would like to retain the use of the spatial reasoning system, since it saves us the effort of writing down a whole set of axioms about spatial relationships. Also, spatial reasoners are often more efficient than generic theorem-proving methods for solving spatial problems (Lindsay 1988). So we would like to treat them as specialized problem-solvers and take advantage of them at appropriate times within a larger reasoning system. So the question is, How can we integrate a spatial reasoning system with a theorem-prover, such that the spatial reasoner takes responsibility for the spatial aspects of a problem, and the theorem-prover handles the non-spatial aspects?

Myers and Konolige (1995) propose an approach to building such a hybrid reasoning system. The two specific reasoning systems they integrate are an analogical subsystem that allows the representation of images with some structural uncertainty, and a sentential subsystem that contains a full first-order inference engine with a variety of methods for deriving new sentences. They interleave processes of *reflection*, where an image is incrementally updated according to derived spatial literals (new facts), and *extraction*, where spatial literals are checked against (or instantiated by) facts that become apparent in the image being formed. They prove that their hybrid procedure is sound, but they argue that it cannot be complete. The incompleteness stems from an inability to reason directly with constraints implicit in the structure of diagrams, which would require reflecting and extracting a broader class of spatial facts from an image, particularly involving disjunction.

In this paper, we propose a new method for integrating a spatial reasoner with a resolution theorem-proving system. Resolution is known to be *refutation complete* for deciding entailment in first-order logic. If a set of formulas is unsatisfiable, for example due to adding the negation of a formula that is entailed by the domain theory, then there is guaranteed to be a derivation of the empty clause by a finite number of resolutions. Unfortunately, in the kind of mixed reasoning problems described above, the domain theory even with the negated query may not be unsatisfiable. A domain theory that contains some spatial information might only be unsatisfiable because there are no images that are consistent with it. For example, consider $left(A, B) \wedge right(A, B)$. The domain theory by itself (without reference to any images or any axioms about spatial relationships) is satisfiable because a model with extensions for these predicates can easily be constructed. However, all of the models that satisfy the theory require some sort of physically unrealizable arrangement. Clearly resolution alone would not be complete in these circumstances.

Our hybrid reasoning approach restores completeness to resolution for these problems by essentially reversing the interaction between the spatial reasoner and the theorem-prover, compared to Myers and Konolige's (1995) approach. Instead of using the spatial reasoner to support logical inference by incrementally building up a concrete model as spatial facts are generated, our approach involves constructing a random image, proving that it contradicts the theory, analyzing the crucial spatial aspects in the proof, using these details to rule out whole classes of similar images, and iterating this process until the space of all possible images is exhausted. Provided that the images are sufficiently well-defined (i.e. have fixed, finite domains), we prove that this algorithm is complete, in that it is guaranteed to detect when a domain theory, even with a mixture of spatial and non-spatial knowledge, is unsatisfiable. This approach is also applicable to integrating resolution with other specialized reasoning systems, such as qualitative reasoners, simulation engines, or other systems with built-in knowledge that can act as consistency constraints among a sub-class of facts within a larger reasoning system. This approach can be used to extend such systems with the ability to reason to outside their domains of expertise.

Reasoning Sub-Systems

Spatial Component

The spatial reasoner we use in this hybrid reasoning system is described in (Ioerger 1994). We begin by defining the formal nature of the images used in the system. Images are essentially array data-structures in which the elements may contain sets of symbols. The arrays are typically two- or three-dimensional, and the resolution of the array (number of divisions per dimension) depends on the application. The arrays represent a region of space by an adjacency-preserving mapping, and the symbols stored in each array element can be used to encode information specific to the corresponding sub-region in space. For example, the location of an object can be indicated by marking the array elements for regions it occupies with a unique label for the object. The space of images is denoted by $\mathcal{I} = (S^n)^d$, where S is the space of (sets of) symbols that can be written into an array element, n is the resolution, and d is the dimensionality.

To describe and ask questions about images, we define a spatial language consisting of terms and predicates. The set of terms T_S, such as Chair-1 or Gear-52, denote objects (unambiguously) in the image. We assume that objects denoted by terms in T_S (and only those objects) occur in every image. We define a set of predicates P_S that express various depictable relationships among the objects. Examples of unary predicates are Green() and Large(). Examples of binary predicates are LeftOf(,), DarkerThan(,), and In(,). The terms and predicates may be combined, along with negations, to form a language of spatial literals L_S, such as ¬LeftOf(Chair-1,Table-2).

The spatial reasoner has two interface procedures, ASSERT and QUERY, that both utilize a core procedure called UPDATE. The spatial reasoner maintains a current working image, which is initialized to a random (legal) image in the space of possible images. Then a sequence of assertions and queries is made. With each assertion of a spatial literal, the system stores the literal and updates the current working image to reflect the new fact along with all of the previous assertions as well as possible. ASSERT uses UPDATE to try to find an alternative of the current working image that satisfies the augmented set of assertions. UPDATE makes small, generic manipulations of an image, such as making minor adjustments in the positions or orientations of objects, or changing their color or size slightly. Among all the possible manipulations for the current image, the one that gives the greatest increase in satisfaction, summed over all the assertions, is selected as the new working image, and this process is iterated until no more improvements can be made. This hill-climbing approach relies on graded interpretations of the spatial relationships, and the tradeoff between its efficiency and its completeness (probability of finding an appropriate image if it exists) can be adjusted by choosing larger or more restricted sets of manipulations (Ioerger 1994).

When a query is made, the system inspects images to determine whether the queried literal is true or false. However, it is possible that the answer might not be exactly as depicted in the current working image. For example, suppose that after ASSERT(Left(A,B)) and ASSERT(Left(C,B)), the working image shows the objects in the following order: A, C, B. Then the answer to QUERY(Left(A,C)) would appear to be YES but should really be UNSPECIFIED. This is an instance of the Indeterminacy Problem (Ioerger 1994), which stems from the use of complete and specific images to represent incomplete spatial information. If some aspect of a situation is not prescribed by a description, then there will be *multiple* images that can represent it, and it is not possible to distinguish between details in any such image that are necessary versus those that are merely consistent. To get around the Indeterminacy Problem, our spatial reasoning system individually tests the consistency of both the queried literal and its negation with the previous assertions. It temporarily asserts the query and uses UPDATE to see if a new image can be generated that is consistent with all the previous assertions. Then it similarly tests the consistency of the negation of the query, and the query can be answered accurately based on which tests succeed in finding reasonable alternative images.

We can characterize the formal role of images in our system in terms of models over the language of spatial literals, which we notate as $M(L_S)$. Treating L_S as a propositional language, a model is simply a truth assignment for all literals in the language. Each image contains a specific arrangement of the objects denoted by T_S, and for each combination of objects, their satisfaction for each of the predicates in P_S can be evaluated. Although fuzzy truth values are often used to represent intermediate degrees of satisfaction for spatial relations, every literal in L_S has a specific truth value in a given image. Thus each image is effectively a model over L_S. However, not all possible models over L_S are legal images. The semantics of predicates such as LeftOf and RightOf are related via their interpretation in images, and the structural constraints of images disallow certain combinations of extensions. Furthermore, the manipulations during the UPDATE procedure may be designed to prevent the generation of "illegal" images that violate certain physical constraints, such as when objects overlap, pass outside of the region, float unsupported in the context of gravity, or change shape. Taken as a whole then, the spatial reasoning system defines a subset of the models over the spatial language: $SR \subset M(L_S)$.

Logical Component

The logical component of our hybrid reasoning system consists of a resolution theorem-prover for first-order logic. We begin by defining a language for the logic that extends the language of spatial literals with some additional (non-spatial) literals, and then adds full first-order syntax. We might also have some additional (non-spatial) terms, and one concern is to prevent the non-spatial terms from appearing as arguments to the original spatial predicates. Let T_N be a set of non-spatial terms and P_N be a set of non-spatial predicates. We can define a language of non-spatial literals L_N from the non-spatial predicates P_N, using terms from either T_S or T_N as arguments. In order to form the atomic formulas for the hybrid language, we combine the spatial literals with the non-spatial literals, also allowing variables V as arguments to either type of predicate (for use with quantifiers):[1]

$$literals ::= P_S \times (T_S \cup V)^* \cup P_N \times (T_N \cup T_S \cup V)^*.$$

The full hybrid language L_H is composed recursively from these literals and the application of the usual quantifiers and connectives. An example of a formula in the hybrid language is: $\forall x \; On(x, Table) \wedge Owns(Pat, Table) \rightarrow Owns(Pat, x)$, where 'On' is a spatial predicate, 'Owns' is a non-spatial predicate, 'Table' is a spatial term, and 'Pat' is a non-spatial term.

Because the hybrid language is an extension of the logical language with some new spatial terms and predicates, models $M(L_H)$ of the hybrid language take the same form as models of any first-order theory: a domain of objects D, denotations that map terms onto objects in the domain, and extensions of predicates specifying which combinations of objects satisfy each predicate. However, we divide the domain into two disjoint subsets: spatial objects D_S and non-spatial objects D_N. We require that the spatial terms T_S denote unique objects in D_S. Furthermore, whereas the extensions of the non-spatial predicates may contain any objects in the entire domain $D = D_S \cup D_N$, the extensions of spatial predicates may contain only spatial objects in D_S.

Models over the hybrid language can easily be projected into models over the spatial language L_S, which is essentially propositional, by converting the extensions of the spatial predicates P_S into an exhaustive list of truth values for every possible combination of objects in D_S that are denoted by some term in T_S. We use the operator P for this *projection* of models: $P : M(L_H) \mapsto M(L_S)$. So a model $m_1 \in M(L_H)$ for a theory in the hybrid language entails a particular model $P(m_1)$ over the spatial language (which might or might not be representable as an image), and conversely, a model $m_2 \in M(L_S)$ over the spatial language, perhaps corresponding to an image, is consistent with a set of extended models $P^{-1}(m_2)$ in the hybrid language.

Inference with a domain theory in the hybrid language may be conducted by resolution (Chang & Lee 1973). First the theory is converted into clausal form by a sequence of well-known truth-preserving operations (DeMorgan's Laws, skolemization, etc.). Then new clauses are derived by the resolution rule from combinations of existing clauses. Schematically, from $P \vee Q$ and $\neg P' \vee R$ we can conclude $Q \vee R$, where the most general substitution needed to unify P and P'

[1] We do not consider function terms in this paper.

is applied to the resulting clause. A common method for using resolution to answer queries is by refutation. If you want to know whether a sentence α is entailed by a knowledge base KB, append its negation to the knowledge base and use resolution to try to derive the empty clause $\Box$ from $KB \cup \{\neg\alpha\}$. If $\Box$ can be derived, then $KB \cup \{\neg\alpha\}$ is unsatisfiable (due to the soundness of resolution), and therefore (assuming the KB itself is satisfiable) α must be entailed by the KB. Herbrand's Theorem states that resolution is complete for refutations; that is, if a knowledge base is unsatisfiable, then there exists a derivation of $\Box$ by a finite number of resolution steps. However, finding such a proof is non-trivial because of the huge number of irrelevant clauses that can be derived. Although many strategies have been proposed for choosing the order in which clauses are resolved to help control the complexity in certain situations, resolution in general is known to be NP-complete (Haken 1985).

Hybrid Reasoning Algorithm

Suppose we are given a domain theory in the hybrid language and a query whose entailment we are to prove. As with standard resolution-refutation theorem provers, the first steps are to add the negation of the queried sentence to the domain theory, and then convert it to clausal form. We will eventually attempt to show that this augmented theory is unsatisfiable, and hence the original sentence is entailed.

Domain Enumeration

The first step that is specific to our hybrid reasoner is called *domain enumeration*. Domain enumeration involves instantiating all variables that occur in spatial literals in the clauses of the domain theory with all possible spatial terms. After the theory has been converted to clausal form, all remaining variables can be assumed to be universally quantified (since existential variables have been skolemized). Thus a predicate that contains a variable as an argument can really be thought of as a schema that stands for many ground predicates with arbitrary instantiations in that argument. In the case of spatial predicates, the model theory for L_H requires extension to contain only objects denoted by spatial terms, of which there are a finite, fixed number. Hence, clauses with spatial literals can simply be expanded by enumerating all possible instantiations via T_S. Each instantiation becomes a new, independent clause. All other occurrences of the same variable within the clause must be substituted identically. If different variables occur in spatial literals within the same clause, they are expanded independently. To illustrate, if R is a spatial predicate and the spatial terms consist of $\{A, B\}$, the domain enumeration of $P(x, C) \vee Q(y, C) \vee R(x, y)$ is:

$$P(A, C) \vee Q(A, C) \vee R(A, A)$$
$$P(A, C) \vee Q(B, C) \vee R(A, B)$$
$$P(B, C) \vee Q(A, C) \vee R(B, A)$$
$$P(B, C) \vee Q(B, C) \vee R(B, B)$$

The initial clause with variables would be replaced by these four new clauses. According to the model theory for the hybrid language, domain enumeration is a truth-preserving operation. The set of models that satisfies the domain-enumerated theory is exactly the set of models that satisfies the original theory.

Another important step in domain enumeration is dealing with Skolem constants. During the process of converting a domain theory into clausal form, one step is to introduce a Skolem constant with a unique name for each existentially quantified variable.[2] In the case of spatial predicates, these arguments must denote one of the objects in the spatial domain. However, a spatial term and a Skolem constant will not automatically unify. Therefore, we must add one extra clause for every Skolem constant appearing in the theory that equates it disjunctively with each of the spatial terms, enforcing a closed-world assumption. For example, consider the sentence $\exists x\ On(x, Table)$. The clausal form of this sentence might be $\{On(Sk1, Table)\}$. If the objects in the spatial domain are known to be {Book,Table,Chair}, then the following clause should be added to the theory: $\{(Sk1{=}Book) \vee (Sk1{=}Table) \vee (Sk1{=}Chair)\}$. An extension of resolution such as paramodulation could be used to handle reasoning with equality in an appropriate manner (Chang & Lee 1973).

Covered List

A central data structure of the hybrid reasoning algorithm is the Covered List (Cov). The Covered List is a disjunctive-normal form (DNF) expression over the language of spatial literals L_S. The Covered List represents *classes of models over L_S that are either illegal as images or are inconsistent with the domain theory.* If a conjunct $\alpha \wedge \beta \wedge \gamma$ occurs in the Covered List, then this means that for any image that mutually satisfies α and β and γ, its model is believed not to be a member of the set of projections of models of the theory. The Covered List is initialized to the empty disjunction.

Main Loop

The main loop of the hybrid reasoning algorithm is given in Figure 1. First, a conjunction of literals (λ) is extracted that falsifies the Covered List. This is done by choosing one literal from each conjunct and negating it. If no consistent set of such literals can be found, then the algorithm halts (discussed below). Once a set of literals has been selected, they are asserted to the spatial reasoner to make a sequence of updates to an initial random image (I). If no consistent image can be found (i.e. the update procedure returns an image that still does not satisfy one of the constraints), then

[2] To maintain a function-free setting, existential variables in spatial literals cannot appear inside the scope of a universal quantifier.

```
let TH be the domain-enumerated, clausal theory
Cov← ∅
while ∃ a conjunction of literals λ s.t. Cov ∧ λ = False
  let I be a random image
  for each literal e in λ, I ← ASSERT(e, I)
  if I still falsifies some literal in λ, Cov ← Cov ∨ λ
  else
    TH' ← ∅
    for each clause c in TH
      if c contains a spatial literal satisfied by I then
        TH' ← TH' ; drop c altogether
      else
        let f be the set of any falsified spatial literals in c
        TH' ← TH' ∪ {(c \ f)_f} ; remove and annotate
    run resolution on TH' until □ is derived
    let CL be the set of clauses in TH' used to derive □
    let σ be the set of spatial literals that appear as
       annotations in CL
    let σ' be the conjunction of negations of literals in σ
    Cov ← Cov ∨ σ'
return "unsatisfiable"
```

Figure 1: The hybrid reasoning algorithm.

the new set of literals is simply added as a conjunct to the Covered List.

However, if an image consistent with the selected literals is found, then we must show that this candidate image is inconsistent with the domain theory. We do this by generating a modification of the domain theory (TH') that incorporates the facts known about this image (called a *specialization* of the theory). Specifically, if a clause contains a spatial literal that holds true in the image, then the whole clause itself will be satisfied. Since a constantly satisfied clause does not help rule out any models, the clause can be dropped from the theory. Any spatial literals in the remaining clauses must then be falsified by the image. If a clause contains a constantly false literal, the literal may be dropped from the clause. These modifications are similar to the Davis-Putnam proof procedure. In the process of dropping the spatial literals from each clause, we annotate the clauses with the literals that are being dropped. This will be important in analyzing the context of proofs.

With this version of the domain theory, specialized for the candidate image, we run a standard resolution inference procedure to try to generate the empty clause.[3] Once the empty clause has been derived, we extract the set of clauses in the adapted theory that were involved in the proof. From the annotations of these clauses, we gather all of the spatial literals (σ) that were falsified by the image, form the conjunction of their negations, and add (disjoin) them to the Covered List. If no set of literals can be found that falsifies this updated Covered List, then the loop terminates, and the algorithm replies that the theory is unsatisfiable.

[3] If the domain theory were satisfiable with respect to the spatial reasoner, then some image would be consistent with it. In this case, the empty clause would not be derivable from the corresponding specialized theory, and resolution might even not halt. But this is irrelevant for refutation proofs. We assume the theory TH is unsatisfiable (with respect to the spatial reasoner) due to adding the negation of an entailed sentence, and we prove in the next section that the specialization of the theory TH' in each iteration entails the empty clause.

Completeness Proof

Our goal in this section is to demonstrate that the hybrid reasoning algorithm above is complete, in the sense that, if the the theory is unsatisfiable, then the algorithm will halt and return this result. By unsatisfiable, we mean that the domain theory might be satisfiable without reference to any images (i.e. there might be a model in L_H), but any models of the theory, when projected into models over the spatial language L_S do not intersect the subset of models for legal images, SR. The formal criterion is: $P(M(TH)) \cap SR = \emptyset$

Theorem: *If the theory is unsatisfiable, then the algorithm in Figure 1 will halt.*

Proof: We begin with an invariant: assume that the models over the spatial language that are covered by the Covered List are either illegal images or are inconsistent with the theory (i.e. do not intersect with any projections of models of the theory). Later we will show that every conjunct we add to the Covered List maintains this property.

The first step in the main loop, illustrated in Figure 2, is to pick a set of literals λ that mutually falsify the Covered List. This set of literals stands for a subset of models $M(\lambda)$ over the spatial language L_S, given arbitrary choices for the remaining literals. None of these models is covered by any conjunct in the Covered List because each of these models has at least one literal with truth value opposite of a literal in each conjunct. The set of literals is completed (i.e. specific choices are made for the remaining literals) by using the spatial reasoner to generate an image consistent with the selected literals. Note that it is possible that the literals in λ are themselves inconsistent, in the sense that they cannot concurrently be true in any actual image. In this case, $M(\lambda) \cap SR = \emptyset$, so adding λ to Cov still maintains the invariant property of ruling out models that are illegal as images or are inconsistent with the theory. If a consistent image can be found, then its model is in $M(\lambda)$ and therefore guaranteed not to be covered by any conjunct in the Covered List.

The middle part of the the main loop of the algorithm deals with specializing the theory with details from the selected image. Recall that, the domain theory by itself might be satisfiable. However, when any satisfying models over the hybrid language L_H are projected onto models over the spatial language L_S, they

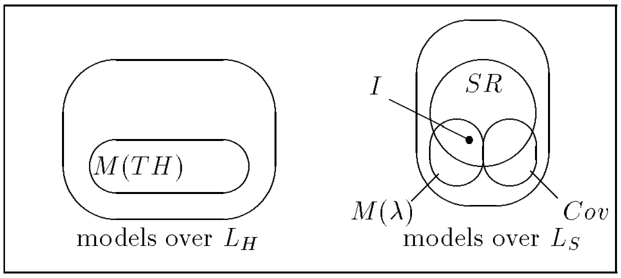

Figure 2: An illustration of the first part of the algorithm, in which a set of literals λ is chosen to be distinct from the Covered List, and then the spatial reasoner is used to find an image I consistent with those literals.

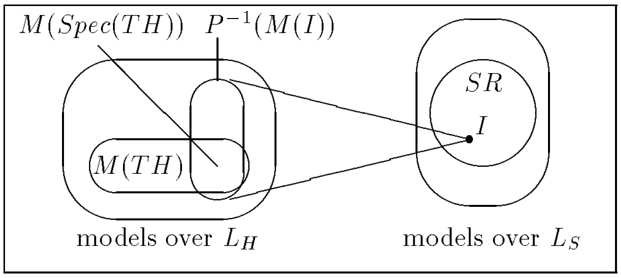

Figure 3: Illustration of the step in which a theory is specialized with details from a selected image. The satisfying models are exactly those in the intersection of the models for the original theory and the models in the inverse projection of the image.

will be found to be illegal images according to the spatial reasoner. So in the context of any particular image, it must be the case that the theory is unsatisfiable. Formally, if we take the inverse projection of the model of the image, this should constitute a set of models in the hybrid language that does not overlap with any models of the domain theory (see Figure 3):

$$M(TH) \cap P^{-1}(M(I)) = \emptyset.$$

We introduce a specialization operator, *Spec*, that takes a domain theory and an image and returns an adapted domain theory that is satisfied by exactly the models in the above intersection: $M(Spec(TH)) = M(TH) \cap P^{-1}(M(I))$. The specialization operator, shown procedurally in Figure 4, is described as follows. For each clause in *TH*, if it contains a spatial literal that is true in the image, then replace that clause with the literal by itself as a unit clause. Otherwise, delete any remaining spatial literals from the clause, which must be false in the image. Finally, for every literal in L_S that is true in the image, add it as a singleton clause to the theory.

Lemma: $M(Spec(TH)) = M(TH) \cap P^{-1}(M(I))$

Proof: First we prove that $M(Spec(TH)) \supset M(TH) \cap P^{-1}(M(I))$. Suppose there is a model in

```
Spec(TH,I)
  TH' ← ∅
  for each clause CL in TH
    if CL contains a literal σ ∈ L_S s.t. Eval(σ,I)=True
      then TH' ← TH' ∪{σ}
    else let Ψ = {σ | σ ∈ CL and σ ∈ L_S}
      TH' ← TH' ∪{CL \ Ψ}
  for each spatial literal σ ∈ L_S s.t. Eval(σ,I)=True
    TH' ← Th' ∪{σ}
  return TH'
```

Figure 4: The theory-specialization algorithm.

the intersection: $m \in M(TH) \cap P^{-1}(M(I))$. Consider the four kinds of clauses in the specialized theory $Spec(TH)$. Some clauses are simply copied over from *TH* since they do not have any spatial literals at all. Clearly they are still satisfied by m since $m \in M(TH)$. Some clauses were replaced by single spatial literals that were made true by the image. Since $m \in P^{-1}(M(I))$ each of these literals is still satisfied, and hence those clauses are satisfied. Some clauses had spatial literals that were dropped because they were falsified by the image. Again, if $m \in P^{-1}(M(I))$, then these literals are still false. So if $m \in M(TH)$, then m satisfies the clause, which means it must satisfy one of the remaining non-spatial literals. So the reduced clause with the spatial literals removed will be satisfied by m. Finally, unit clauses were added for all spatial literals that were satisfied by the image. Each of these will also be satisfied in the inverse projection of the model of the image. So m satisfies all clauses in $Spec(TH)$, and thus $m \in M(Spec(TH))$.

Now we prove containment in the opposite direction: $M(Spec(TH)) \subset M(TH) \cap P^{-1}(M(I))$. Consider a model of the specialization of the theory, $m \in M(Spec(TH))$. Each clause in *TH* has a reduced counterpart in $Spec(TH)$, that is, a clause with a subset of literals. If m satisfies $Spec(TH)$, then it satisfies each of these clauses. And if m satisfies a given clause, then it satisfies all supersets of that clause. So $m \in M(TH)$. Also, $Spec(TH)$ contains unit clauses for all the spatial literals that were true in the image. So $m \in P^{-1}(M(I))$. Thus we can conclude that $m \in M(TH) \cap P^{-1}(M(I))$.

We have shown that, with this specialization operator, we can create a modified domain theory that satisfies the following models: $M(TH) \cap P^{-1}(M(I))$. We have also shown that, because the theory is supposed to be unsatisfiable in the context of any image, $M(TH) \cap P^{-1}(M(I)) = \emptyset$. Hence the specialized theory should be unsatisfiable: $M(Spec(TH)) = \emptyset$. Now resolution can be applied, and we are guaranteed of the existence of a finite derivation of $\Box$. At the very least, an enumeration of proofs in order of length is guaranteed to eventually find a derivation of $\Box$.

The hybrid reasoning algorithm does not exactly compute the specialization of the domain theory with respect to the selected image. It leaves out all of the unit clauses with spatial literals for efficiency (avoids having to generate a complete description of the image). This should not affect the existence of a derivation of the empty clause because none of these clauses with spatial literals could be involved in the proof. In the specialized theory, either a spatial literal $\sigma \in L_S$ appears or its negation $\neg\sigma \in L_S$ appears, but not both (since all literals that are false in the image are deleted and all literals that are true are added to the theory). Thus a singleton clause with a spatial literal can never be resolved with another clause because the negation of it does not occur in any other clause. So a derivation of $\Box$ can still be constructed without these clauses.

Continuing with the completeness proof for the hybrid reasoning algorithm, the next step after deriving the empty clause by resolution is to collect the clauses from the specialized theory that were used in the proof. Some of these clauses originally contained spatial literals that were falsified by the selected image; the literals were removed from the clause and annotated to it. We can now use them to identify a context in which the same proof (derivation of $\Box$) would succeed again. Specifically, in any other image in which the same literals are falsified, the specialization of the domain theory would again be unsatisfiable. Therefore, any image that satisfies the conjunction of the negations of these literals will be inconsistent with the theory. The final step in the loop is to add this conjunction to the Covered List, maintaining its invariant property.

One last point finishes the completeness proof: every new conjunct added to the Covered List covers at least one new model over the language of spatial literals. The reason for this is that the conjunction of literals is guaranteed to be satisfied by the image constructed in the current iteration because they are negations of (annotated) literals that were falsified by the image. We showed earlier that the model for this image was guaranteed to not be covered by any conjunct in the Covered List. So this new conjunct necessarily covers at least one new model. Because the language of spatial literals L_S is finite, and hence the space of models over it $M(L_S)$ is finite, the monotonic growth of the Covered List guarantees that it will eventually cover the entire space, at which point the algorithm will halt and reply that the original domain theory is unsatisfiable.

Discussion

Hybrid Reasoning

There are various hybrid reasoning approaches for integrating a theorem-proving system with other kinds of problem-solvers. One class of hybrid reasoners treats the problem-solver as a kind of *specialized inference step* that can be used to combine sentences that would not otherwise be allowed. An example of this approach is theory resolution (Stickel 1985), which extends unification in domain-specific ways. Sorted logics (Walther 1988) and terminological reasoners (Brachman, Gilbert, & Levesque 1985) can also be looked at in this perspective, since they allow certain new inferences based on external information, such as about types. Van Baalen (1991) explores the use of theory resolution for incorporating specialized satisfaction-testing procedures, including some that are common to spatial reasoning. A second class of hybrid reasoners includes those that take a *model-based* approach, in which a set of facts about a specific situation is incrementally built-up and used to check the consistency of other sentences during theorem-proving. Constraint-logic programming (Jaffar & Lassez 1987) is a well-known example of this approach. Myers and Konolige's (1995) hybrid system, with its accumulation of facts about an image via the reflection process, instantiates this model-based approach for spatial reasoning.

In contrast, our hybrid reasoning method is more like an *intelligent case-based analysis*. Because each image makes a certain part of models of the theory concrete, it forms a context in which the consequences of the domain theory can be worked out conditionally. If that image fails to accommodate the theory, then other images can be considered. But because the number of possible images is finite, eventually the unsatisfiability of a theory will be detected. This reversal of the interaction between the theorem-prover and the spatial reasoner is analogous to Burckert's (1994) hybrid reasoning approach, in which constraints are collected during resolution and attached as a context to empty clauses that are derived. A refutation proof is complete when enough empty clauses have been generated that the disjunction of all of their sets of residual constraints covers all possible assignments.

A brute-force method that simply tests a theory in the context of all possible images would be very inefficient. Our algorithm avoids this inefficiency by analyzing the proof of the empty clause in each iteration of the algorithm to extract the subset of spatial literals that were critical to the derivation. Usually there will be only a few such literals involved, and these constitute a generalization that can be used to rule out a whole class of similar images (when added as a conjunct to the Covered List). Subsequent images are specifically chosen to avoid these same conditions. Thus, while in the worst case, the algorithm is only guaranteed to rule out one model at a time (total of $2^{|L_S|/2}$), it will often halt after only a few iterations.

Spatial Uncertainty

For the purposes of building a hybrid reasoning system, a crucial requirement of the spatial reasoner is the ability to handle *indeterminacy*. Images are characteristically complete and specific, making it difficult for them to encode incomplete descriptions or partial information. Yet interacting with a theorem-prover often involves reasoning about isolated parts of a scene, perhaps leaving other aspects entirely unspecified. In fact, the theory may involve explicit disjunction about some spatial relationships. The approach that Myers and Konolige (1995) take is to encode some *structural uncertainty* in images, which can be used to depict a degree of ambiguity in certain details.

The spatial reasoner in this paper, based on the one described in (Ioerger 1994), uses *dynamic manipulation* of images to handle indeterminacy, essentially generating consistent alternatives as needed. Given an initial set of constraints, a representative image is constructed that best satisfies the constraints, effectively extending an incomplete description into a concrete instantiation. Then, if this image turns out to be inadequate (e.g. inconsistent with additional constraints), an acceptable alternative is sought by making a sequence of small adjustments, such as in the position of objects. Each of these two approaches is appropriate in different circumstances. One requires designing structural uncertainty in image representations, and the other requires designing image manipulation operators.

Limitations and Extensions

There are some limitations to the hybrid reasoning method we have proposed in this paper. The first is that it requires the domain of spatial objects to be known *a priori*. A fixed, finite set of objects is needed for domain enumeration. While this requirement facilitates concrete reasoning with images, and is easily satisfied in many domains, there may be applications where solving hybrid-spatial problems involves uncertainty about the existence of objects themselves in the image. In such cases, a more flexible approach to domain enumeration, such as in (Myers & Konolige 1995) may be needed. A second limitation is that domain enumeration can potentially cause a great increase in the number of clauses. If a clause contains a spatial literal with a variable, then it is replaced by D times as many clauses, where D is the number of objects in the image. If the clause contains n independent variables occurring in spatial literals, then D^n clauses must be added. Such an expansion of the size of the domain theory might make resolution much less efficient. Solving these issues remains the focus of future work.

Our hybrid reasoning approach can also be applied to integrating other kinds of reasoners with resolution. It can be used with any reasoning system that: 1) has a discrete representation (at least at one level) for understanding constraints in propositional form, and 2) can find consistent completions for partial specifications. For example, our approach might be used to integrate a qualitative model of a system with a resolution theorem-prover by using simulation or envisionment to derive possible responses of some variables, given specifications for others. Or a neural net could be integrated with a logical system by using a Boltzmann procedure (Ackley, Hinton, & Sejnowski 1985) to find consistent states of the network where some inputs are clamped and others are allowed to float freely.

References

Ackley, D.; Hinton, G.; and Sejnowski, T. 1985. A learning algorithm for Boltzmann machines. *Cognitive Science* 9:147–169.

Brachman, R.; Gilbert, V.; and Levesque, H. 1985. An essential hybrid reasoning system: Knowledge and symbol level accounts of KRYPTON. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 532–539.

Burckert, H. 1994. A resolution principle for constrained logics. *Artificial Intelligence* 66:235–271.

Chang, C., and Lee, R. 1973. *Symbolic Logic and Mechanical Theorem Proving*. New York: Academic Press.

Funt, B. V. 1980. Problem-solving with diagrammatic representations. *Artificial Intelligence* 13:201–230.

Glasgow, J.; Narayanan, N.; and Chandrasekaran, B., eds. 1995. *Diagrammatic Reasoning*. Menlo Park, CA: AAAI Press.

Haken, A. 1985. The intractability of resolution. *Theoretical Computer Science* 39:297–308.

Ioerger, T. 1994. The manipulation of images to handle the indeterminacy problem in spatial reasoning. *Cognitive Science* 18:551–593.

Jaffar, J., and Lassez, J. 1987. Constraint-logic programming. In *Proceedings of the 14th ACM Principles of Programming Languages Conference*, 111–119.

Kaufmann, S. G. 1991. A formal theory of spatial reasoning. In *Principles of Knowledge Representation and Reasoning, Proc. Second Intl. Conf.*, 347–356.

Lindsay, R. K. 1988. Images and inference. *Cognition* 29:229–250.

Mukerjee, A., and Joe, G. 1990. A qualitative model for space. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 721–727.

Myers, K., and Konolige, K. 1995. Reasoning with analogical representations. In *Diagrammatic Reasoning*. Menlo Park, CA: AAAI Press. 273–301.

Papadias, D., and Glasgow, J. 1991. A knowledge-representation scheme for computational imagery. In *Proceedings of the Thirteenth Annual Conference on of the Cognitive Science Society*, 49–54.

Stickel, M. 1985. Automated deduction by theory resolution. *Journal of Automated Reasoning* 1:333–355.

VanBaalen, J. 1991. The completeness of DRAT, a technique for automatic design of satisfiability procedures. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, 514–525.

Walther, C. 1988. Many-sorted unification. *Journal of the ACM* 35:1–17.