Artificial Intelligence 175 (2011) 299–345

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

How does a box work? A study in the qualitative dynamics of solid objects ☆

Ernest Davis

Dept. of Computer Science, New York University, United States

ARTICLE INFO

Article history:
Available online 3 April 2010

Keywords:
Qualitative physics
Naive physics
Commonsense reasoning
Solid objects
Kinematics
Dynamics
Planning

ABSTRACT

This paper is an in-depth study of qualitative physical reasoning about one particular scenario: using a box to carry a collection of objects from one place to another. Specifically we consider the plan, `plan1` "Load objects `uCargo` into box `oBox` one by one; carry `oBox` from location `l1` to location `l2`". We present qualitative constraints on the shape, starting position, and material properties of `uCargo` and `oBox` and on the characteristics of the motion that suffice to make it virtually certain that `plan1` can be successfully executed. We develop a theory, consisting mostly of first-order statements together with two default rules, that supports an inference of the form "If conditions XYZ hold, and the agent attempts to carry out `plan1` then presumably he will succeed". Our theory is elaboration tolerant in the sense that carrying out the analogous inference for carrying objects in boxes with lids, in boxes with small holes, or on trays can reuse much of the same knowledge. The theory integrates reasoning about continuous time, Euclidean space, commonsense dynamics of solid objects, and semantics of partially specified plans.

1. Introduction

How does a box work? You won't find an explanation in *The Way Things Work* [21] or at HowStuffWorks.com, for the very good reason that any fool can *see* without assistance how a box works; and if you can't see it, an elaborate explanation won't help. Indeed, the question, "How does a box work?" seems almost ill-formed; it violates the Gricean condition that a question must admit a useful answer.

That people do indeed *understand*, in a productive and general sense, how a box works[1] is evidenced by the fact that they can reason about how the functionality of a box relates to the geometric and physical properties of the box itself and the objects it is used with. They understand, for example, that objects cannot come out of a closed box; that objects can be carried in an open box, if the box is moved smoothly and held upright; that objects will fall out of an open box if it is turned upside down; that more and larger objects will fit in a large box than in a small box; and so on. These inferences can be carried out using only qualitative information about the geometry of the boxes and objects involved; precise specifications of the geometry and material properties is not required. The knowledge involved is not specific to boxes; much the same knowledge is used in the commonsense understanding of trays, shelves, drawers, and so on.

☆ My thanks to David Stewart, for information about the current state of the art in scientific computation in the theory of rigid solid objects; to Steve LaValle, for discussions about the relations of this work to robotics; to Leora Morgenstern and Vladimir Lifschitz, for discussions of non-monotonic inference; and to the reviewers for helpful suggestions. This research was supported in part by NSF grants IIS-097537 and IIS-0534809.

E-mail address: davise@cs.nyu.edu.

[1] As opposed to, for instance, a theory that posits that the people's interactions with boxes can be characterized in terms of stimulus/response behavior that subjects learn from positive and negative reinforcement when they put objects into boxes and take them out.

0004-3702/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.artint.2010.04.006

Table 1
PDDL theory of boxes.

```
(define (domain Box)
(:types object location)
(:predicates   (at ?o - object ?l - location)
               (in ?o ?b - object)
)

(:action load
 :parameters   (?o ?b - object ?l - location)
 :precondition (and (at ?o ?l) (at ?b ?l) (box ?b))
 :effect       (and (in ?o ?b) (not (at ?o ?l))))

(:action move
 :parameters   (?o - object ?l1 ?l2 - location)
 :precondition (at ?o ?l1)
 :effect       (and (at ?o ?l2) (not (at ?o ?l1))))

(:action unload
 :parameters   (?o ?b - object ?l - location)
 :precondition (and (in ?o ?b) (at ?b ?l)
 :effect       (and (at ?o ?l) (not (in ?o ?b))))
)
```

But, of course, what is obvious to people can be very difficult to make obvious to computers. Currently there exist implementable theories at two levels that can be applied to boxes. On the one hand, there is the exact theory of rigid solid objects: given exact geometrical and material specifications of the box, of the objects inside, and of the motion of the box, one can calculate exactly the resulting motions of the objects.[2] On the other hand, one can develop a discrete, abstract representation of the domain of boxes, using fluents like "in(O, B)" and actions like "putin(O, B)" and "move($B, L1, L2$)". (Table 1 shows one such representation in PDDL [28].)

What is lacking is a theory at an intermediate level; a theory that, on the one hand explicitly deals with the geometry of the objects and motions involved, and, on the other hand, allows those geometries to be specified partially or qualitatively. Clearly it is often important to be able to reason at such a level; exact geometric characteristics of objects may be unknown, or one may wish to reason about classes of situations generically, or one may wish to reason about a system at an early stage of design, before the exact geometry has been decided on. Clearly, also, human commonsense reasoning is often able to deal with such reasoning without difficulty. Our objective in this paper is to develop a representation and a theory of moving objects in boxes at this level.

This paper is thus a contribution to a small corner of the research programme first pioneered by John McCarthy nearly fifty years ago [22,23]. In the very earliest days of AI, McCarthy foresaw that the representation of commonsense domains and the automation of commonsense knowledge would be one of the major challenges in constructing intelligent programs; and that formal logic would be a powerful tool in constructing well-defined representations and powerful inference techniques. The years that have passed since then have entirely confirmed McCarthy's original insights. Though short-term and small-scale progress in AI has often been made through the deliberate and systematic avoidance of issues of commonsense reasoning (see e.g. [5], pp. xvii–xviii), it has become ever clearer that, in the long run, no general intelligence can be achieved without dealing with these issues, and that using representational systems that lack a well-defined logical semantics yields no real advantages and inevitably leads to muddle [26]. The research in this paper has also been deeply influenced by other aspects of McCarthy's work, particularly his study of non-monotonic inference [24] as a critical feature of commonsense reasoning, and his advocacy of elaboration tolerance [25] as a desideratum in the development of domain theories.

The other major inspiration for the research in this paper is Pat Hayes' "Naive Physics Manifesto" [17], which advocated carrying out McCarthy's representation project in the particular area of commonsense physical reasoning.

The direct practical applications of a qualitative theory of boxes — e.g. for household or industrial robots that deal with boxes; for deep understanding of natural language texts that describe the use of boxes; or for interpretation of video showing manipulation of boxes — would hardly in itself justify, in terms of a cost-benefit analysis, my labors of writing this paper, your labors of reading it, and the labors, not yet begun, of implementing it and integrating it with such an application [7]. Boxes arise too rarely in these applications; it would be much more cost-effective either to use one of the levels of representation and reasoning that currently exist, to contrive some application-specific hack, or to live with the slight gap in functionality entailed in an imperfect understanding of boxes.

Rather, the importance of the theory we develop here is as part of a general theory of qualitative physical reasoning. There is good reason to hope that large parts of the conceptual analysis, the representation, and the formal theory can be carried over to a more general theory of qualitative physical reasoning; that our experience in developing the theory of

[2] This is actually more problematic than one might suppose, as we will discuss in Section 2 below.

boxes will be helpful in designing such a theory; and that a general theory, when complete or nearly so, will be so powerful and broadly applicable as to justify the very large costs of development [19].

This paper will deal with one specific, basic use of boxes: an open box can be used to carry a collection of objects, the *cargo*, from one place to another place. More specifically, we are concerned with the following plan (henceforth `plan1`)

Load the objects one by one into the box;
Move the box to its destination.[3]

In this paper, we will formulate a set of qualitative boundary conditions on the box, the cargo, and the initial state sufficient to support the inference that `plan1` will execute successfully. We will present a formal theory in which this inference can be carried out; this theory integrates continuous time, Euclidean space, physical dynamics of solid objects, and the semantics of partially specified plans. We will give an extensive sketch of the formal proof of the correctness of the plan.

The key characteristic of our analysis is that the verification is achieved using only qualitative geometric and physical specifications. Neither the number nor the shape of the cargo objects is constrained and the shape of the box is constrained only by the requirement that it is a box and is substantially larger than the combined size of the cargo. We do not have to assume that other objects do not exist,[4] only that they do not directly interfere with the loading and the carrying. The trajectories used in loading the blocks and in carrying the box are a little more constrained, but they too are permitted a large measure of freedom. Furthermore, we do not require that the objects remain in the position in which they are released; they may topple over or shift around, either when they are originally released, or when some other object is loaded on top of them, or while the box is being carted around. However, though they shift, they remain in the box; this is one of the main functions of a box. Therefore the planner does not have to make sure that he loads the objects in a stable position.

To ensure that our theory is elaboration tolerant [25] to a reasonable degree, and is not narrowly confined to this one specific problem, we have kept a number of variant problems in mind. The theory is designed to extend fairly easily to support the following variant inferences and scenarios:

1. Infer that, if the box is turned upside down and held that way while being carried, the objects will fall out, and the plan will fail.
2. Infer that, if one or more of the cargo objects is attached to the ground then they cannot be loaded into the box.
3. Infer that, if the box is attached to the ground, then it cannot be carried to the destination.
4. Infer that, if the trajectory of the box is bumpy enough, then cargo objects may be thrown out and the plan will fail.
5. Infer that, if object O is placed inside box $B1$ which is then placed in box $B2$, and $B2$ is carried to $L2$, then both O and $B1$ will come along with it.
6. Infer that, if a lid is placed on the box after it is loaded and kept there while it is carried, then the objects will stay in the box regardless of the motion of the box.
7. Infer that, if a lid is placed on the box before it is completely loaded, then the loading cannot be completed.
8. Infer that objects may be placed on an open tray and carried from one location to another, but that they cannot be piled as high or moved as roughly as in a box.
9. Infer that objects can be carried in a box with holes in the bottom or sides, like a milk crate as long as the objects are too large to fall through the holes.

Due to limitations of time and space, we have not carried out a complete formal analysis of any of these variants, but we are quite confident that the theory presented here can be extended without substantial difficulty to cover all or most of these. Specifically, inferences 1, 4, and 8 would certainly require additional physical axioms. The extensions needed for 1 and 4 should be straightforward; inference 8 is substantially more challenging. Note that, if one uses an open tray as in inference 8, objects must be stacked stably and maintain a fixed position, or one risks their falling off the side of the tray. The other inferences do not require any new physical axioms; these can all be carried out within the current theory with at most the addition of some additional geometric definitions and lemmas. Similarly, we have designed our theory so that it is not inconsistent with the standard Newtonian theory of solid object dynamics (see Section 2.1), with the idea that the two theories can be merged in future work.

One particular objective in this paper is to formulate the physical knowledge used in terms that avoid or minimize the use of differential equations and forces, which are central to the Newtonian theory. As discussed at length in [6], analysis in terms of forces and of behavior over differential time is particularly unsuited to qualitative reasoning in this domain, because many scenarios which can be simply characterized over extended time are both extremely complicated and extremely unstable when analyzed over differential time. Consider dropping an object on the ground and watching it settle to a stable state. The characterization of its behavior in differential terms, between the time it first hit the ground

[3] We had originally hoped to include a final step of unloading the objects one by one; but it turns out that formulating conditions that guarantee that it is possible to unload the cargo raises new and difficult problems. We hope to return to this in future work.

[4] Readers who are not KR researchers may be surprised that this is worth mentioning; but such strong "closed world" assumptions are in fact ubiquitous in automated commonsense reasoning and planning.

and the time it comes to rest, can be very complicated; the thing rotates, spins, slides, bounces while impacts and forces come and go. It is also very unstable; the exact sequence of impacts, forces, slidings, and so on depends very delicately and discontinuously on the exact shapes and material properties and the initial conditions. The characterization of its behavior over extended time is very simple and robust; within a few seconds, it is at rest on the ground, not far from the initial point of impact. As we shall see, our theory achieves this objective to a very large degree; in fact, the physical theory we need for this problem makes no reference to velocities, accelerations, or forces.

Let us make clear at the outset a few objectives that this paper does *not* attempt to achieve:

- We assume a single agent. The semantics is not easily extended to a world with multiple agents.
- It does not cover all cases in which it is commonsensically obvious that the plan works.
- It does not derive the rules from “first principles” of Newtonian physics, for reasons that will be discussed below.
- It does not support reasoning about likelihood, or relative likelihood. We hope to address this in future work.
- It does not discuss how this reasoning could be implemented in practice.

The paper is structured as follows: Section 2 reviews related work, in mathematical physics (Section 2.1), in AI qualitative physical reasoning (Section 2.2), and in robotics (Section 2.3). Section 3 gives a pre-formal analysis of `plan1`, discusses the many ways in which the plan can fail, and presents ways to formulate the boundary conditions, the physical constraints, and the plan itself so that the plan can be relied on to succeed. Section 4 shows how this domain theory can be expressed in a formal first-order theory. Section 5 sketches a proof of the correctness of `plan1`; a complete proof is given in a Web-based appendix at http://cs.nyu.edu/faculty/davise/box-proof.pdf. Section 6 summarizes our results and discusses future work.

2. Related work

Previous work relevant to the research described here falls into three categories: mathematical physics analyzing the Newtonian dynamics of rigid solid objects (Section 2.1), work in AI on qualitative dynamics of rigid solid objects (Section 2.2), and work on robotic planning (Section 2.3).

2.1. The Newtonian theory of rigid solid objects

The study of the dynamics of solid objects, idealized as perfectly rigid, was begun by Galileo and Newton and continued by physicists and mathematicians of the eighteenth, nineteenth, and twentieth centuries. It has recently enjoyed a revival of research interest because of its many applications, which include physical simulation, robotics, computer-aided manufacturing, animation, virtual reality, and video games. Recent research includes modelling issues (how best to model the interactions of rigid objects), computational issues (how to effectively compute the behavior of a system of rigid objects), and theoretical issues (showing that every well-posed boundary value problem has a solution). Nonetheless, there remain many fundamental unsolved problems in this domain. Stewart [36] surveys the literature and discusses the state of the art.

In the *kinematic theory* of rigid solid objects, an object O is characterized by its *shape*, which is the region of space that O occupies in some standard position. We will assume throughout this paper that the shape of an object is bounded, regular (equal to the closure of its interior), and has a connected interior. In this section, though not in the remainder of the paper, we will further assume that the boundary is smooth; that is, there is a unique tangent plane at every boundary point. The position of object O at time T is determined by a rigid (orthonormal) mapping, called the *placement* of O at T. The region that O occupies at time T, called the *place* of O at T is equal to the image of the shape of O under the placement of O at T.

The kinematic theory consists of three constraints:

1. Each object maintains a fixed shape. This is guaranteed by the above constraint that the place of O at T is related to its shape by a rigid mapping.
2. The placement of O at time T is a continuous function of T.
3. If $O1 \neq O2$ then the places of $O1$ and $O2$ at T do not overlap.

The kinematic theory is unproblematic and well-understood.

The *dynamic theory* extends the kinematic theory in the following ways: An object is further characterized by a density distribution over its shape, its coefficient of friction against other objects, and its elasticity. For each object O, at each time T_1, the limit of derivative of the placement of O at T as T approaches T_1 from below, and the limit as T approaches T_1 from above, both exist, though they are not necessarily equal. We will call these the velocity *before* T_1 and *after* T_1 respectively. Two objects $O1$ and $O2$ that are in contact at a given time may interact in one of two ways: First, they may exert a *force* on one another, distributed over the region of contact. Alternatively, when two objects collide, they may exchange a finite quantity of momentum instantaneously through exerting an *impulse* force on one another.

In addition to forces between objects in contact, there are *external forces*, particularly the earth’s gravity and forces that result from the actions of autonomous agents. In most problems, some objects are specified to be *fixed;* that is, they do not move under any circumstances.

The effect of forces and impulse forces on non-fixed objects is given by generalizations of Newton's second law:

4.A. $\vec{F} = M\,d\vec{v}/dt$, where $\vec{F}$ is the net force on an object O, M is the mass of O, and $\vec{v}$ is the linear velocity of the center of mass of O.
4.B. $\vec{J} = M\Delta\vec{v}$, where J is the net impulse forces acting on object O, and $\Delta\vec{v}$ is the discontinuous change in the linear velocity; i.e. the difference between the linear velocity of the center of mass after T minus the linear velocity of the center of mass before T.
4.C. $\vec{T} = I\,d\vec{\omega}/dt$, where $\vec{T}$ is the net torque, I is the tensor of inertia, and $\vec{\omega}$ is the angular velocity.
4.D. $\vec{W} = I\Delta\vec{\omega}$ where $\vec{W}$ is the net impulse torque and $\Delta\vec{\omega}$ is the discontinuous change in the angular velocity.

Newton's third law asserts that:

5. The force (ordinary force or impulse force) exerted by $O1$ on $O2$ at point P is exactly equal to the negative of the force exerted by $O2$ on $O1$ at P.

Finally, contact forces between objects are generated in the following ways: (The generation of external forces lies outside this theory; the external forces are given as boundary conditions in a problem.)

6. (Constraint forces.) Two objects in contact may exert *constraint* forces on one another. The direction of constraint force from $O1$ on $O2$ at contact point P is normal to their common tangent at P and points out of $O1$ into $O2$. The magnitude of the constraint force is just large enough to ensure that, when all the forces are combined, the non-overlapping condition is maintained.
7. (Coulomb friction.) Suppose that $O1$ and $O2$ are in contact at point P; there exists a non-zero constraint force $\vec{N}$ between $O1$ and $O2$ at P; and the surfaces of $O1$ and $O2$ are moving at velocity $\vec{v}$ relative to one another at P. If $\vec{v} \neq \vec{0}$, then there is a *sliding frictive force* from $O1$ to $O2$ whose direction is $-\vec{v}$ and whose magnitude is $\mu_k|\vec{N}|$, where μ_k is the kinetic coefficient of friction between the material of $O1$ and the material of $O2$. If $\vec{v} = \vec{0}$, there is a *static frictive force* between $O1$ and $O2$ whose magnitude is at most $\mu_s|\vec{N}|$ and whose value is such that the equations of motion have a solution. The coefficient μ_s is the static coefficient of friction.
8. If $O1$ and $O2$ are in contact at point P at time $T1$ and their velocities before $T1$ would cause them to interpenetrate in the neighborhood P, this is a *collision* at P. The result of a collision is that $O1$ and $O2$ exert an *impulse* force on one another directed along the normal to their common tangent. There are a number of different models for determining the magnitude of this force; Newton's "experimental law" is often used but is sometimes problematic. All of them involve a real-valued parameter, the *coefficient of restitution,* which depends on the material properties of $O1$ and $O2$.
9. If $O1$ and $O2$ collide at time T, and they are part of a set of objects S which are spatially connected at T, then the impulse force between $O1$ and $O2$ may propagate to the other objects in S (e.g. a cue billiard ball hitting into a set of other balls, or a croquet mallet ($O1$) striking one ball $O2$ and knocking away a second ball). It is not well-established how best to model this propagated impulse.
10. If $O1$ and $O2$ collide at point P at time $T1$ and their surfaces have a relative velocity at P which has a non-zero component $\vec{v}_t$ in the common tangent plane, then the two objects exert an impulse force on each other in the direction $-\vec{v}_t$. This is seen, for instance, in the way that the spin of a ball may change at a collision. Again, it is not well-established how to model this interaction.
11. Under some circumstances, the equations of motion only admit a solution if there is a impulse force between two non-colliding objects. See D. Stewart's solution to the Painlevé paradox [35,36].

The formalization of this theory is further complicated by the fact that two objects may be in contact at a finite collection of isolated points, at every point on a curve, at every point on a surface, or at the union of a surface, a curve, and isolated points, and that the place and form of contact can change discontinuously over time.

There are a number of unresolved issues in formulating this theory. Ideally, one would like to have a theory that

(a) agrees well with experimental measurements, up to the limits of the idealization;
(b) has at least one solution for every well-posed boundary-value problem;
(c) is demonstrably equal to the limit of the theory of elastic solids, as the elasticity goes to zero;
(d) is demonstrably the limit of numerical computation, as the precision increases and the time-step goes to zero.

But this ideal has not yet been attained. Strong results, specifically the existence of a solution for well-posed boundary-value problems, have been proved only under restricted conditions. There are a number of different sets of conditions that have been proved sufficient to guarantee the existence of a solution; the following is typical:

- Two objects are only in contact at a finite collection of points, not over an extended region.
- The normals at the contact points between two objects lie in a single hemisphere.
- Collisions are inelastic.

There is also a small philosophical/logical literature on axiomatizing the physics of rigid solid objects. The interest here is mostly on issues in theoretical physics and philosophy of science, rather than on detailed physical models. For example, the axiomatization presented by Adams [1] includes only items 4.A and 4.C above; it does not even include the constraint that objects do not overlap.

The needs of knowledge-based commonsense reasoning are rather different from those of scientific computation. In a theory of commonsense reasoning, coverage is more important than precision; it is better to be able to rule out grossly impossible behaviors in all situations than to be able to give precise answers over a limited class of situations.

Let me conclude here with some general comments about the relation between the scientific theory of solid objects and commonsense knowledge of the same domain. The scientific theory intended to deal with every possible case of a collection of objects sliding, spinning, rolling, and colliding. However, the problems that arise in everyday life are almost always much more constrained in one respect or another. Indeed, there is no reason to think that commonsense understanding is very good at dealing with the general case; naive subjects find as basic a phenomenon as gyroscopic motion baffling and hard to believe even when directly experiencing it.

In ordinary situations, people are usually interested in maintaining a very large measure of control over the objects they interact with; and it is hard to achieve control over an object that is moving freely.[5] Also, a free motion of an object is likely to end in a collision; and people avoid subjecting objects to collisions for fear of damage. Commonsense reasoning is thus primarily concerned with cases where objects either stay where they are without need for intervention, or with cases where an agent moves an object in a controlled way. On the other hand, a commonsense theory must to some extent take account of uncontrolled free motion, if only to be able to allow some useful predictions in the cases where this happens by accident.

In physics, the base case of the theory is an object moving with constant velocity under Newton's first law without external forces.[6] In commonsense reasoning, the base case is an object sitting motionless on a table.

2.2. *Qualitative physical reasoning*

This project continues the work on qualitative reasoning about solid objects reported in [6], which presented a logical analysis of the inference that a marble dropped inside a funnel would fall out the bottom of the funnel. Rule-based approaches to commonsense physical reasoning ultimately derive from [17]. Bennett et al. [3] presents a purely geometric theory of rigid object kinematics in a language entirely defined in terms of the primitives "Region $R1$ is a part of region $R2$", and "Region R is a sphere".

Some of the fundamental difficulties and limitations of this methodology are discussed in [7]. A current survey of work in AI physical reasoning may be found in [10].

Previous work on qualitative reasoning about kinematics includes [12,29]. Forbus et al. [14] extend these to include dynamic analyses of certain kinds. Gelsey [15] reports a simulator for predicting the dynamic interactions of solid objects. All of these require an exact shape description of the objects involved to be input. De Kleer's NEWTON program [11] and Forbus's FROB program [13] carried out qualitative prediction of the behavior of point objects interacting with fixed constraints whose shape is qualitatively described. Stahovich et al. [34] present a system for qualitative analysis of a limited class of dynamic systems; this is similar to [14] but more elegant and more clearly defined, though more limited in scope. They claim that their system can work from a rough sketch of the objects involved, but it is not clear how this works.

2.3. *Robotics*

The literature on robotics discusses many of the same issues as are addressed here, but from a sufficiently different angle that the techniques applied there are rarely directly applicable here. (LaValle's [18] recent textbook is an extensive and excellent survey of robotic planning.) We will discuss briefly a couple of issues that these two lines of research have in common; a more extensive comparison is beyond the scope of this paper.

Analysis of the mechanics of manipulation. As we will discuss below, in this paper we limit ourselves to an extremely simplified model, in which a disembodied agent moves one object at a time through telekinesis, but in a broader setting, commonsense knowledge is aware of and reasons about physical aspects of manipulation, and we hope to address these in future work. Even in this setting, however, it seems likely that there is a divergence between the roboticists' analysis and the analysis needed for commonsense reasoning. Roboticists must carry their analysis to the level needed to actually execute the manipulations involved, whereas it would seem that commonsense reasoning stops at a more abstract level, and leaves the ultimate implementation in muscular forces to learned control patterns. On the other hand, robotics research

[5] The most common contexts involving freely moving objects are aircraft and spacecraft; military and hunting projectiles, from slingshots and arrows to guns and grenades; and sports that involve balls, pucks, shuttlecocks and so on. I think it is safe to say that, with the exceptions of sports, dropping trash into a basket, and perhaps tossing bags of laundry, most contemporary Americans rarely deliberately toss or drop any large solid objects. I should be interested to learn of any further exceptions. Note, by contrast, that liquids and collections of small solid objects like salt or coffee are generally poured.

[6] The rotational motion of such an object can be surprisingly complicated.

tends to focus on limited classes of controlled situations; for commonsense reasoning, it is important to reason about what an agent can effectuate in any circumstance.

Information limited planning. A very interesting branch of recent robotics research studies how a robot with limited knowledge of the environment can nonetheless plan to achieve specified goals (see [18], Chaps. 11 and 12). These studies obviously have elements in common with qualitative reasoning about planning; both deal with constructing plans in situations that are not completely specified. But there is, I think, an important distinction centering around the standpoint of the reasoning being done. The information space analysis in robotics takes a first-person approach: the agent who is reasoning is the robot who is acting, and he has to get enough information to be able to actually carry out the actions involved. Qualitative reasoning takes a third-person approach: the reasoning is being done by someone other than the agent himself who has partial knowledge of the situation and wants to be able to reason that an agent, who may himself be omniscient or who may have limited knowledge, would be able to carry out a partially specified plan.

Direction of inference. Finally, both robotics planning research and AI planning research focuses almost exclusively on constructing and executing plans to meet specified goals. In commonsense reasoning this is only one of many possible reasoning tasks. Other directions of inference include determining that a given goal cannot be achieved; inferring characteristics of the environment or the agent from the execution of a plan, or from the failed attempt to execute a plan, and so on. All these draw on the same kind of knowledge as plan construction, and therefore a general domain theory should support all of them and a general knowledge-based reasoner should be able to carry all of them out.

3. The execution of `plan1`: pre-formal analysis

Our central objective in this paper is to validate `plan1`; that is, to show that, under suitable conditions, `plan1` is a reasonable plan and can be expected to succeed in achieving the goal of moving a collection objects to a destination. The hard part of this analysis is actually at the pre-formal[7] level: deciding which issues should be addressed in detail, which should be idealized, and which should be ignored; what problems should be covered; what knowledge must be used; and what assumptions must be made. Once all this is determined, the translation of this analysis into logical notation is, as we shall see, comparatively straightforward.

We will assume throughout this paper that objects can be idealized as rigid and solid; thus, we do not have to worry about the box breaking or the objects becoming crushed.

3.1. A model of manipulating an object

An execution of `plan1` consists primarily of a sequence of manipulations of individual objects. We therefore start our analysis of the plan by formulating a theory of manipulation.

What kinds of manipulations are in fact possible for a given physical agent depends on the geometry of its manipulators and the geometrical and physical constraints that govern them. For real agents, animal or robotic, these tend to be complicated. In order to abstract and simplify the details of the manipulator, we will use instead the following idealized model of manipulation: we conceptualize the agent as, so to speak, having telekinetic powers over one object at a time. That is, the agent may choose any object OM and may move it along any physically possible path; we need not specify how the physical manipulators of the robot could actually reach, grasp, and move the object to accomplish this. The motion of the manipulated object OM may cause motions of other objects, either because of kinematic constraints or because of frictive forces. If other objects are in the way of the attempted motion of OM then OM will exert a force on them. If this force causes the other objects to move out of the way, then the motion is possible; if not, the motion is impossible.

The effect of the idealization is to abstract away the robot's actual manipulators. For instance, as compared to the capacities of a human hand or an anthropomorphic robotic hand, this idealization allows us to ignore such issues as what positions of the fingers and palm relative to the object allow the object to be grasped; the space occupied by the hand; the constraints on motion placed by the structure of the hand and arm; and the limits on the strength of the hand. For the most part, therefore, the idealization is more powerful than a real manipulator. There is one thing, however, that a real manipulator can do that this idealization cannot; namely, to directly move more than one object at a time. A human hand can hold several small objects simultaneously, but our idealization does not allow this. (We could, of course, change the idealization to specify that the agent can move arbitrarily many objects within reach simultaneously. However, this is undesirable, because an agent who could do this has no use for boxes; it can simply carry all the cargo directly.) In future work, we hope to consider more realistic models of manipulation.

[7] We do not mean to suggest that methodologically, "pre-formal analysis" is or should be completed before formalization begins. On the contrary, the process of constructing formal axiomatizations of domain knowledge and formal proofs gives important insights into the inherent nature of the knowledge involved. In practice, research proceeds on all three fronts concurrently.

3.2. Characteristics of plan1

Let us begin by writing out plan1 in more detail. Let oBox be the box, uCargo be the set of cargo objects, oTable1 be the surface initially holding the objects, and oTable2 be the surface to which we wish to move the loaded box. Then we define plan1 in pseudo-code as follows

```
plan1 ≡
{ while (not all objects in uCargo are inside oBox)
        { $O1$ := some accessible object uCargo outside oBox;
          load $O1$ into oBox;
        }
  move oBox from oTable1 to oTable2;
}
```

Three characteristics of plan1 will be critical for our analysis. First, plan1 is hugely underspecified. In particular, a complete implementation using our idealized robot would have to specify how the plan executor is to choose the next cargo object to move, the position within (or above) the box at which to release each object, and the trajectory along which to move each object and the box.

Second, plan1 is very well suited to reasoning with qualitative information. It makes no *a priori* assumptions about the number, shapes, material properties, or initial positions of the cargo objects involved. Neither the agent himself nor an external reasoner needs to have this information to see that this is a reasonable plan; indeed, the plan is often executable by an agent who never gets this information. For instance, an agent with no visual or other precise spatial perception and with only an approximate idea of the position of its own manipulator may well be able to execute the plan by groping for the objects, getting them inside the box, and releasing them.

The third characteristic is not a feature but a problem; as we elaborate below in Section 3.3, there are a large number of ways in which the plan can fail. If we wish to posit conditions that allow the conclusion "The plan is guaranteed to succeed" to be inferred with certainty, then we must necessarily posit quite restrictive conditions on the spatial and physical characteristics of the box and the cargo, and we may also have to impose greater specificity in the plan statement. That is, we have to negotiate a three-way trade-off between

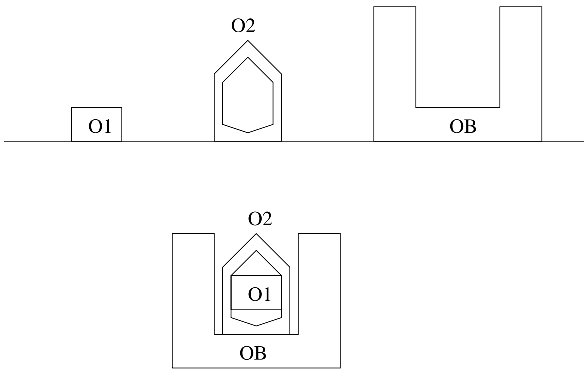
(a) adding further specifications to the plan;
(b) requiring that the objects and box meet more restrictive conditions;
(c) allowing the conclusions we draw to be plausible or likely but uncertain.

The trade-off between constraining the class of objects and specifying the plan amounts to the observation that, by using more intelligence about how to carry out the plan, one can apply it successfully to a substantially greater class of objects. This is not surprising; packing a box efficiently is, after all, an enterprise that requires some thought and skill, and is not achieved by tossing objects at random into the box. The trade-off between degree of certainty and the other two categories amounts to the observation that plans carried out haphazardly or in borderline circumstances are more likely to go wrong; again, this is not surprising. In this paper we will explore a small number of ways in which these trade-offs can be made. There are many different combinations of conditions and plan specifications that, to a commonsense understanding, justify the conclusion that the plan [necessarily/probably/possibly] will achieve the goal; the theory developed in this paper will cover only a small fraction of these, though of course a complete theory commonsense theory of boxes would cover all of them.

Establishing that plan1 is a valid plan to achieve the goal "The objects in uCargo are above oTable2" starting in state s1 involves that showing that every step of plan1 will be executable at its proper time and that at the end of any execution of plan1, the objects in uCargo are all above oTable2. Specifically, we need to posit or establish the following propositions:

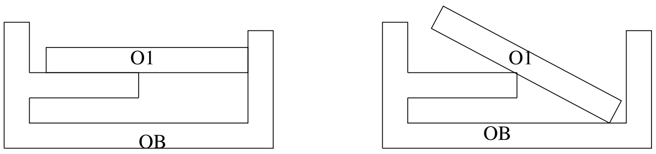
1. In s1, the cargo uCargo and the box oBox are all on oTable1.
2. At each iteration of the loading loop, there is some object $O1$ in uCargo that can be loaded into oBox.
3. After $O1$ has been loaded into oBox, $O1$ will be in oBox.
4. No cargo object exits oBox while $O1$ is loaded into oBox.
5. After the completion of the loading loop, it is feasible to carry oBox from oTable1 to oTable2.
6. No objects come out of oBox while it is being carried from oTable1 to oTable2.
7. After oBox has been carried from oTable1 to oTable2, oBox is on oTable2.
8. If oBox is on oTable2 and $O1$ is in oBox then $O1$ is above oTable2.

Clearly if the above propositions are true then plan1 is executable and succeeds in bringing the cargo uCargo to oTable2. However, these propositions are not formulated in a directly usable form. In particular, as far as possible, we want our problem to be formulated in such a way that we posit that state s1 satisfies a collection of specified conditions



$O1$ and $O2$ will fit inside OB only if $O1$ is teleported to the inner cavity of $O2$.

Fig. 1. Bug 3.A.



$O1$ can be placed inside OB but falls over so that it is partly outside.

Fig. 2. Bug 3.B.

and that the plan is executed starting in `s1`; and then from these boundary conditions we infer the correctness of conditions such as 2–8 above that characterize the world at later times. Our analysis below achieves this objective in large measure but not completely. In particular, we need to posit *isolation conditions* that objects other than the cargo, the tables, and the box do not interfere with the execution of the plan. (We could replace these with initial conditions on the external objects which support the isolation conditions as an inference; however, it seemed to us that positing the weak isolation conditions is more reasonable for commonsense reasoning than positing the much stronger initial conditions on external objects that would be required.)

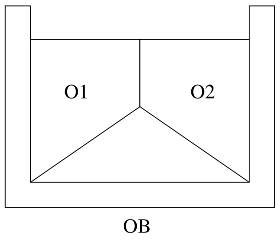
At the outset, let us assume some conditions that are obviously necessary or useful:

COND.1 In state `s1`, object `oBox` and cargo `uCargo` are stably supported on `oTable1`.
COND.2 There is a "protected region" `manipSpace1` which contains `uCargo` and `oBox` and is large enough so that each object can be moved through `manipSpace1` into `oBox`; and it is guaranteed that no objects except `uCargo`, `oBox`, and `oTable2` ever enter into `manipSpace1`.
COND.3 The cargo `uCargo` fits inside `oBox`. That is, there is a configuration C in which each of the objects in `uCargo` is in the inside of `oBox` and which is physically feasible in the sense that no two overlap.
COND.4 No other object reaches into the inside of `oBox` while it is being carried. (Condition COND.2 guarantees that this does not happen during loading.)
COND.5 `oBox` can be fully on top of `oTable2` in a stable position.
COND.6 There is a second "protected region" `manipSpace2` which contains the starting and ending positions of oBox and which is large enough that `oBox` can be moved through `manipSpace2` from start to end while remaining upright. No object other than `oBox` and `uCargo` enter into `manipSpace2` during the execution of `plan1`.

3.3. Potential problems and their fixes

Given conditions COND.1–COND.6 above, what can go wrong with the plan? Actually, it can go wrong in more ways than one might at first suppose. (For convenience of cross-reference, I include here with each bug a forward pointer to the discussion of how the bug is addressed.)

BUG.1 `oTable1` itself interferes with the loading of some object $O1$ into the box, or with carrying of `oBox` away from L1. E.g. $O1$ or `oBox` is fastened to `oTable1`; or `oTable1` encloses $O1$ in a cage. (Addressed in Section 3.3.1.)
BUG.2 In the process of loading $O1$ into the box, it may knock against one of the other cargo objects which then falls out of the protected region `manipSpace1` and gets trapped by some other object, so that it can no longer be loaded. (Addressed in Section 3.3.1.)
BUG.3 The target configuration C in which `uCargo` fits inside `oBox` may be unattainable, for any of a number of reasons:



Whichever of $O1, O2$ is loaded first will fall over so as to block the loading of the other.

Fig. 3. Bug 3.C.

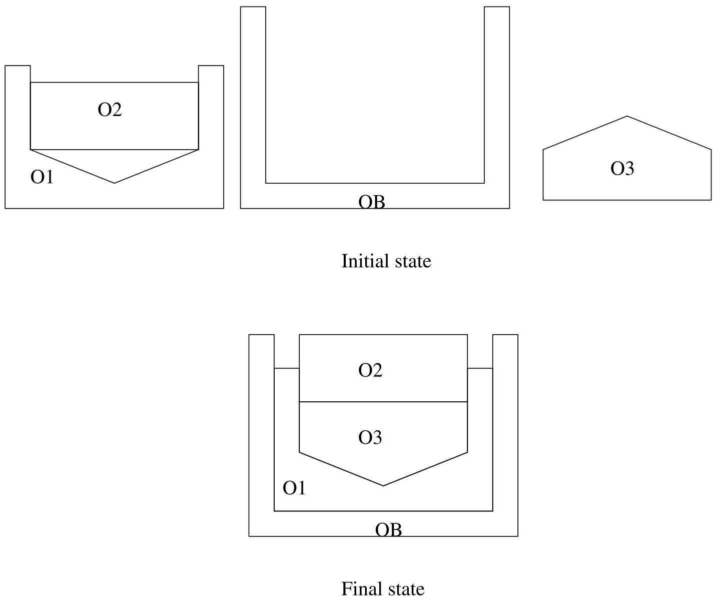


Fig. 4. Bug 4.

a. It may be kinematically unattainable; e.g. it may depend on teleporting one object in `uCargo` into an inner cavity of another object, or disconnecting two objects that are fastened together (Fig. 1).
b. It may be unstable; that is, if the objects are placed in C, they may fall into a different position where they are not entirely inside the box (Fig. 2).
c. It may be dynamically unattainable; that is, it cannot be attained by an agent who can only manipulate one object at a time and who is loading one object at a time (Fig. 3).

(All addressed in Section 3.3.2.)

BUG.4 It is possible that in loading $O1$ into the box, $O2$ will be carried along with it, and will end up in a position which precludes completing the loading of the box. E.g. $O1$ is a box and initially $O2$ is inside $O1$, but the cargo can only be fit into OB if some other object $O3$ is placed inside $O1$ underneath $O2$. The plan `plan1` does not include any method for dealing with this, even if, intuitively, it is easily solved by moving $O2$ either before or after $O1$ has been loaded (Fig. 4). (Addressed in Section 3.3.1.)

BUG.5 Loading a cargo object into the box or lifting the box may cause a trap to be sprung (as in the opening scene of *Raiders of the Lost Ark*), which prevents the completion of the plan. This can happen either when the cargo object is lifted, when it is placed in the box, or when the box is lifted. (Addressed in Section 3.3.1.)

BUG.6 The box may fall over during loading, or it may be knocked off the table. (Addressed in Section 3.3.6.)

BUG.7 Suppose that, at some stage of plan execution, object $O1$ is sitting on the long end of a lever OL, and a heavy object $O2$ is dropped or falls onto the short end of OL. Then $O1$ can be catapulted far from where it is supposed to be (Fig. 5). Specifically,

a. During loading, if $O1$ and OL are inside the box, and $O2$ is placed on the other end of the lever, $O1$ can be catapulted outside the box. (Addressed in Section 3.3.5.)
b. During loading, if $O1$ and OL are outside the box, $O2$ is sitting on the other end of the lever, and $O1$'s end of the lever is being held down by another heavy object $O3$, then lifting $O3$ to load it may cause the lever to be released, catapulting $O1$ out of reach. (Addressed in Section 3.3.1.)

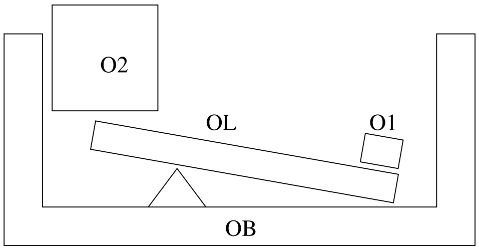


Fig. 5. Bug 7.

c. During carrying, if $O1$ is on lever OL inside the box, and the other objects inside the box shift in one way or another, then $O2$ may fall onto the other end of the lever, catapulting $O1$ out of the box. (Addressed in Section 3.3.5.)

There are also other, even more far-fetched, scenarios in which an object can fly out of the box; for example, a tiddlywinks effect; or a wedge positioned in a crack between two heavy objects may be shot upward if the two objects are squeezed together; or an elastic object like a basketball lying quietly on the ground may bounce upward if it is hit sharply from the top. We will consider all such possibilities as coming into the category of BUG.7.

BUG.8 Objects may come out of the box while it is carried from `oTable1` to `oTable2`.

a. If the box leans over too far, the cargo may fall out. (Addressed in Section 3.3.3.)
b. Similarly, if the box first leans to the left, and then to the right, then the cargo objects may accumulate on the left-hand wall and then be lifted up about the opening. If the box leans back and forth numerous times, then the cargo can gradually climb the walls, being first carried up with each rising wall and then settling down into a higher position on the opposite wall. (Addressed in Section 3.3.5.)
c. If the motion of the box is very violent then the objects may be flung out. (Addressed in Section 3.3.4.)
d. If the objects are very elastic and the motion is bumpy they may bounce out. (Imagine carrying a wooden box full of ping-pong balls.) (Addressed in Section 3.3.1.)
e. If the objects can roll or slide with little friction, and the inside of the box is curved (like a bowl) then they can build up a “sloshing” resonant motion inside the box that eventually allows them to escape, even if the motion of the box is quite smooth. (Addressed in Section 3.3.1.)
f. If the sides of the box are slanted out, and the motion of the box is bumpy, then a cargo object can be gradually bumped up the back side of the box and eventually out. (Addressed in Section 3.3.4.)
g. If the sides of the box are slanted out, and the box is spun rapidly, then centrifugal force may pull the objects out. (Addressed in Section 3.3.4.)
h. If there are many cargo objects (e.g. a heap of sand), then in the course of motion the heap of objects can shape itself into a ramp or bowl, allowing an object to escape methods (d), (e), or (h), even if the sides of the box itself are vertical. (Addressed in Section 3.3.5.)

With eight categories of bugs enumerated (plus subcategories), one might well wonder whether this is just the beginning of a list that can be extended indefinitely. As it happens, once we have addressed these bugs, we will have a plan that can be formally validated, so this list of bugs is complete in the sense that it suggests a complete set of fixes.

It should be noted that, with the significant exceptions of bugs 5, 7, 8.b, 8.f, 8.g, and 8.h, all of these are familiar to anyone who has extensively moved objects in boxes (that is to say, pretty much everyone), and people take them into account in planning, executing, and correcting their box moving activities. (The same is true of the additional bugs with variant boxes to be discussed in Section 3.4.) These are not at all unusual, and the strategies and conditions that we will suggest to eliminate them are likewise formalized versions of commonsense understanding.

We need to exclude all these bugs one way or another before we can validate the plan. As discussed above, there are three general categories of fixes:

1. Imposing stronger conditions on the shapes, material properties, initial configurations, and exogenous motions of the objects involved. Such conditions reduces the scope of the correctness proof, and require that the agent have more complete knowledge of the state. This approach also leads to conditions on the plans that are more restrictive than actually necessary. For instance, we will impose much stronger conditions than are actually necessary on the relation between the size of the cargo and the size of the box, in order to be able to be able to prove the general statement that whenever these conditions are met, the objects can be loaded into the box (see Section 3.3.2).
2. Making the plan more specific, so that the agent uses some intelligence and care about how he carries out the loading, carrying, and unloading. This makes more demands on the agent; it also generally requires a more complex analysis in generating the correctness proof.
3. Achieving less certainty in the conclusions; concluding that the plan is probably correct or correct by default rather than that it is guaranteed to be correct.

3.3.1. *Basic geometric and physical conditions*

To begin with, we will posit some fairly stringent geometric and physical conditions. First, since our primary interest in this paper is in the use of the box, not in the theory of disassembling heaps, we will assume that initially the objects are all placed separately on the fixed support `oTable1`, and that each object has a clear space above it, so that it can be lifted vertically upward without touching any other object, and that it in fact is lifted without touching any other object. Second, we will assume that the objects in `uCargo` cannot roll, have a high coefficient of friction, and a low coefficient of restitution (that is, they don't bounce).

These two assumptions, between them, eliminate BUG.1 (that `oTable1` blocks the loading of some object), BUG.2 (loading one object knocks some other cargo object out of the way), BUG.4 (that loading $O1$ brings $O2$ along in some interfering way), BUG.5 (that loading an object or lifting the box will trigger a trap), BUG.7.b (that loading one object results in an object that is outside the box being catapulted), BUG.8.d (bouncing objects out of the box), and BUG.8.e (objects sloshing out of the box).

3.3.2. *Fitting the cargo in the box*

BUG.4, that the fitting configuration is unattainable, can be dealt with by requiring `oBox` to be much larger than `uCargo`. We consider two particular kinds of conditions here, corresponding to different specifications of `plan1`. If we want `plan1` to allow the agent to place the objects anywhere at all that they fit inside the box, then we can apply the following condition: Let `maxDiam` be the maximum diameter of any object in uCargo. Let `rDeep` be the region of all points in the X–Y plane where the inside of `oBox` has at least depth `maxDiam` and let `nSquare` be the number of non-overlapping squares of side `maxDiam` in `rDeep`. Then, as long the number of cargo objects is less than `nSquare`/4, they can be loaded into the box in this way, since any cargo object already in the box can block at most 4 such squares.

The above result is not very satisfying since this does not allow any more cargo objects to be loaded in a deep box than in a shallow one, as long as the depth is greater than `maxDiam`. Unfortunately, that is pretty much unavoidable with such a weakly constrained loading strategy. The problem is that a sufficiently perverse and ingenious agent may be able to use the first objects in `uCargo` to build a *dome* just inside the top of the box, which will make it impossible to add more cargo no matter how large the volume underneath the dome. Note that our correctness proof is supposed to establish that the plan succeeds however the agent chooses to execute it as long as he follows the constraints defined in the plan; we are not allowed to complain that the agent is following the letter but not the spirit of what we had in mind.

The following loading strategy is more effective: the agent is required to load each cargo object to a reasonably low open position — not necessarily to the lowest possible position, just not to a position that is entirely higher than some other option. Specifically, the agent is prohibited from loading a cargo object to position $M1$ inside the box if it is possible to load it instead to some other position $M2$, such that highest point of $M2$ is lower than the lowest point of $M1$. If this not very restrictive protocol is followed, then the following can easily be shown. Suppose that inside oBox there is an empty rectangular cuboid of dimensions `lCube` long by `wCube` wide by `hCube` high. Then as long as the number of cargo objects is less than

$$\lfloor \mathtt{lCube}\,/2\,\mathtt{maxDiam}\rfloor \cdot \lfloor \mathtt{wCube}\,/2\,\mathtt{maxDiam}\rfloor \cdot \lfloor \mathtt{hCube}\,/2\,\mathtt{maxDiam}\rfloor$$

this loading strategy will get all the cargo into the box. In our formal axiomatization and correctness proof we will use this plan and this constraint.

This analysis, too, is far from optimal; it generally overestimates the space needed in the box by a factor of 8, and makes no allowance for the efficient stacking of large numbers of long thin objects. Since the focus of this paper is not optimal packing, we have pursued this no further here. Readers who want to try their hands at getting stronger results, however, should note that, in the context of the theory in this paper, merely proving a geometric result, that there is a configuration in which all the cargo objects fit within the box, is not sufficient because it risks running into BUG.3. You have to show that, if you load the objects into the box, they will remain in or settle into a position in which they all fit. "Remaining in" is a problem in qualitative statics; "settle into" is a problem in qualitative dynamics. For instance, if you are loading a set of pencils into a case, you need to show that they do not somehow arrange themselves into a complicated lattice work that blocks further loading while leaving plenty of empty space within the box. Our proof of the correctness of the strategy above does indeed allow the possibility that the objects settle within the box after they have been placed, but we demonstrate that even if they do settle, the loading can be completed.

3.3.3. *Tipping out*

Bug 8.a, that the box is tilted so far in carrying that some of the cargo falls out, can be eliminated for the case of a single tilting motion (i.e. without a gradual climbing of the walls, as in 8.b) by requiring that the object not be loaded too close to the top of the box and that the box be kept close to vertical. Assume that, if any objects in the box settles into a new position as a result of tilting the box, its center of mass moves downward in setting. Let O be any cargo object, and let F be the diameter of O. Let H be the maximum horizontal distance between a point inside the box and a point in the top of the box in the initial state; let D be the minimum vertical gap between the top of the box and the center of mass of O in the initial state; and let ϕ be the maximum tilt of the box from the vertical orientation. If $D\cos(\phi) > H\sin(\phi) + F$ then O cannot be tilted out of the box (Fig. 6). (This additional gap of F is necessary to block the possibility of some curved object

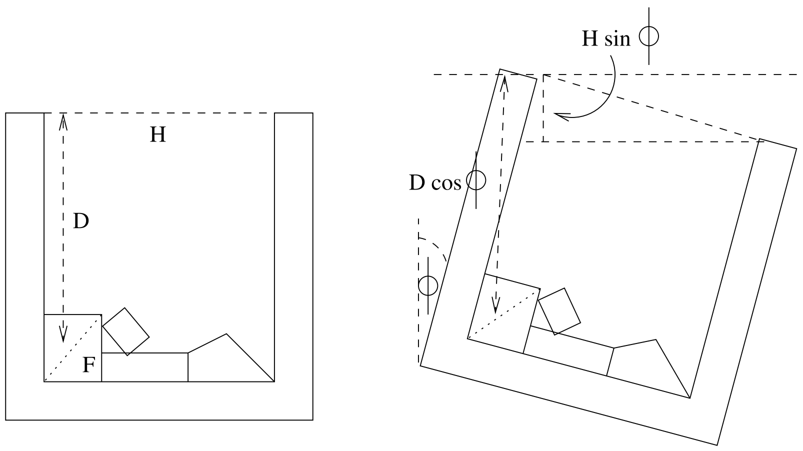


Fig. 6. Tilting out of the box.

executing a "Fosbury flop" in which the object can escape from the box while keeping the center of mass below the top of the box.)

3.3.4. Smoothness of motion

Bugs 7.b (that cargo objects are flung out of the box), 8.f (that the objects work their way up the back of the box in a series of small bumps), and 8.g (that the objects are spun out of the box) could be eliminated by requiring that (a) the cargo objects are not loaded too near the top of the box; (b) motions of the box moved smoothly and not spun rapidly; and (c) the sides of the box are steep.

In fact, we have not included these conditions in our formal analysis; rather, for simplicity we have relied on the default rule, introduced in the next section, that objects in a box generally do not move higher with respect to the box. However, let us briefly discuss the nature of these inferences in the more complete theory we eventually hope to attain.

Commonsensically, there is a multi-way tradeoff between the bumpiness and rotational velocity of the motion of the box, the steepness of the sides of the box, the gap between the top of the objects and the top of the box, and the likelihood that objects will come out of the box — one may as well throw in here the elasticity of the objects as well. However, this tradeoff

- involves elements that are hard to quantify, such as the "bumpiness" of a trajectory;
- would be very hard to justify by a formal analysis based on Newtonian mechanics, even if one confines attention to cases where the probability is 1 or 0;
- is almost certainly not known very precisely by actual commonsense (human) reasoners.

What reasoners do know, it seems reasonable to say, is a number of specific cases drawn from experience, and the general structure of the tradeoff. This enables them to conclude that, for example, if the box is shaken up and down very violently, the objects will certainly come out the top; that if you fill a box to the brim with small objects and then roll it in a wheelbarrow down a bumpy road, it is quite likely the objects will come out; and that if the cargo objects are inelastic and you carry the box carefully and smoothly then the objects will certainly stay in the box. We hope to address these issues in future work, but they are beyond the scope of the current paper.

3.3.5. Catapulting

BUG.7.a and BUG.7.c, that an object may be catapulted out of the box,[8] are much more problematic than those we have considered above. Two features of this bug are immediately apparent. First, it rarely if ever actually happens. My guess is that no one in this history of packing boxes has ever been surprised by an object flying out in this way.[9] Second, it obviously could happen if the agent specifically sets it up to happen. All that is needed is an object to act as fulcrum, an object to act as lever (or a single bent object to act as both fulcrum and lever), a light object to act as missile, and a heavy object (much heavier than both missile and lever) to act as trigger (Fig. 5). Further thought reveals a third feature; namely, that it is very difficult to formulate plausible constraints on the shapes of the objects or on the loading actions that suffice to make this impossible. The fulcrum and missile need only be comparatively small; the shape of the trigger is entirely unconstrained. The lever does have to be substantially longer than it is thick, but ruling out all objects of that kind — that is, requiring

[8] We have already dealt with BUG.7.b in which an object on `oTable1` is catapulted out of the protected area.

[9] One of the reviewers disagrees, and thinks that probably this does occasionally happen in practice. Also, note, by contrast, that liquid spilling out of the top of a pail, either in filling the pail or in carrying it (analogous to BUG.8) is quite common and requires care to avoid if the pail is nearly full.

that all objects be more or less cubical — really does seem like an unacceptably strong restriction to fix a problem that never actually comes up. Note that the trigger need not be the object being loaded, so it does not suffice to posit that the agent sets each object down very gently in loading it; adding the new object, however gently, may disturb a carefully balanced equilibrium and knock over some *other* object that will function as the trigger. One might think that one could find restrictions on the way in which objects are heaped and removed from heaps that would exclude this, but short of doing an exact simulation, which defeats the whole point of qualitative reasoning, that seems very difficult to do.

So what should be done? The obvious temptation is just to posit that it can't happen, since it never does. But I am afraid that this is a dangerous path. Quite clearly this *can* happen. It is not merely a theoretical consequence of Newtonian physics, or a recondite case with specially designed objects; on the contrary it is quite obvious that anyone can make it happen in a few minutes with a couple of household objects. Thus, a theory in which it cannot happen is inconsistent with commonsense understanding of solid objects, and therefore cannot be extended to a general theory of commonsense reasoning about solid objects. Note that this is quite different from, for example, saying that the theory of rigid solid objects cannot be extended to a theory of non-rigid objects. The latter case has to do with theories of two very different scopes; naturally, one does not expect to deal with non-rigid objects in formulating a theory of rigid objects. Here the natural scope of the overarching theory is the dynamic theory of rigid objects, which includes both boxes and catapults. In this paper, we happen to be dealing with boxes, but certainly we would certainly hope to be able to develop our theory as a subset of a broader theory of rigid solid objects.

A second idea is to restrict the physics to quasi-statics, the limiting case where dissipative forces are always so large as compared to momentum that no object can travel more than a negligible distance under its own inertia; they only move when pushed by an agent or by gravity. Usually quasi-static theories are used in the context of two-dimensional objects moving on a horizontal surface, but one could develop such a theory for three-dimensional motion; intuitively, you imagine the entire scenario as taking place in a vat of Liquid Prell.[10] But we have decided not to pursue this. First, this extension to three-dimensional dynamics involves some technical difficulties (for instance, how fast should an object move in free-fall or in falling over while partially supported?) Second, making this assumption obviously limits the generality of the theory.

A third idea is to move to a probabilistic theory, especially as we will eventually have to do that anyway, and say that the catapulting is very improbable. However, as far as I can tell, this does not fix the underlying problem, though it changes it. The problem now is that a probabilistic theory must specify the probability of events, not just absolutely, but conditionally as well. And the probability of catapulting, which is negligibly small under most conditions, becomes 1 on the condition that the agent deliberately sets up a catapult. One might be tempted to say that it is very unlikely that the agent will *want* to set up a catapult, but at that point we would be basing our physical theory on a theory of relative likelihood of goals, which does not seem like a good direction to go in.

The best solution I have been able to find is to use a default theory. Intuitively, we want to posit a default rule that catapulting does not occur unless there is good reason to think that it might. Thus if we know that the agent has set up a catapult and is triggering it, then the catapulting must occur, the theory predicts that it will occur, and the default does not apply. If there is "good reason" to think that the agent may have set up and triggered a catapult, then the conditions of the default rule are negated, and we are left agnostic as to whether the catapulting occurs. Otherwise, the default rule applies, and we conclude that no catapulting occurs.

Specifically, we define an "upward-motion-free" history as one in which no object in a heap ever moves upwards with respect to the object(s) supporting the heap. We then posit a default rule that histories are, by default, upward-motion-free. The default rule does not apply to cases where an agent picks up the object directly or indirectly. (Some care must be taken in cases where the heap is supported by several objects or the support tilts in the course of the history. We will give a more exact statement in Section 3.7 after defining "heap" in Section 3.6.)

This default rule also takes care of all the subcategories of BUG.8 except BUG.8.a, since they all rely on an object escaping out the top of the box. This is indeed how we will deal with these bugs in our formal theory. A more complete theory would contain rules that would specify exceptional cases in which these bugs are likely or certain to occur; given such rules, one could then exclude the bugs for ordinary cases, either by using default rules or by explicitly negating their enabling conditions.

Generally, the major difficulty in using a default rule of the form "Assume P unless P is impossible" is making sure that P is not impossible; that is, P is not contradicted by anything else in the theory. For qualitative prediction, the following approach is possible: Suppose that you are given qualitative information about the boundary conditions; that is, the shapes and positions of the objects and the actions of the manipulator. If you can find a specific instance I that satisfies the qualitative constraint, such that when you run a Newtonian simulator on I, the defaults are satisfied, then clearly the defaults are consistent with the boundary conditions and with Newtonian physics, and may therefore be applied.

Of course, going from a monotonic theory to a non-monotonic theory has many drawbacks. For our purposes, the most serious drawback is that the value of establishing any particular inference shrinks dramatically, in the following sense: In a monotonic theory, if you prove that $\Gamma \models \phi$, then you have simultaneously established that $\Gamma \cup \Delta \models \phi$ for any further axioms Δ. Thus, you have established the validity of inferring ϕ from a whole class of possible knowledge bases. By contrast,

[10] A brand of shampoo that, some years ago, was the subject of a well-known advertising campaign featuring a demonstration that the shampoo was particularly viscous. For some reason, this was supposed to imply that it was also particularly good shampoo.

in non-monotonic theories this does not follow so easily; for any particular Δ you have to re-establish that Δ does not block any of the non-monotonic rules that you used to prove ϕ from Γ. In probabilistic theories, the situation is even more difficult; you have to show that ϕ is conditionally independent of Δ given Γ, or approximately independent.

A second general problem with non-monotonic theories is that it is very hard to be confident that they don't have unintended consequences. You can check that a monotonic theory is safe by constructing a model in which it is true; then the consequences of the theory cannot be any weirder than the model. But non-monotonic theories do not have semantics of this kind. (This is one reason that recent work on non-monotonic theory has tended to use the non-monotonic inference to derive equivalent monotonic axioms; these can then be checked for validity in a model.) To alleviate this problem, we try to keep our default rules as weak as possible, consistent with supporting our desired inference that objects ordinarily remain in boxes during loading and carrying, so that there is as little risk as possible that the default rules have unintended consequences.

Another problem is the Yale Shooting Problem [16]. Our default rule does indeed run into this; it allows a backwards causality in which objects may be catapulted out at an earlier stage if that will prevent them from being catapulted out later. However, since this only comes up in scenarios of Rube Goldberg-like complexity, I am not very concerned about it. I don't think there's much point worrying about how to fix this until one has addressed the underlying probabilistic issues.

It may seem odd that I should now be worrying about the Yale Shooting Problem at all since this problem was "solved" many years ago. The difficulty is that all the solutions I know of (e.g. [20,33]) work by using closed-world assumptions to derive a suitable frame axiom, in the manner mentioned above. If that would work here, I wouldn't need a default rule at all; I could just state the frame axiom monotonically. (The authors of these solutions are working with different underlying constraints as to what constitutes an acceptable formulation of a prediction problem.)

Finally, it might be argued that the proposed default rule is just a way of disguising the argument, which we raised and rejected above, that catapulting is unusual because agents rarely have the goal of firing catapults. After all, if the population of agents involved consisted entirely of seven-year olds who had just discovered the joys of catapulting things, then it might be reasonable to assume, by default, that whenever a plan can be instantiated so as to fire a catapult, it will be. Thus, our proposed defaults above incorporate an assumption about the psychology of the agents involved. I don't think this is right. It seems to me that there are two separate plausible inferences involved here; first, that the physical inference that loading objects into a box randomly will rarely cause catapulting, and second, the psychological inference that agents engaged in loading a box will rarely decide to construct a catapult. A physical theory of boxes must deal with the first, independently of the second.

3.3.6. The box falls over

There are at least four different scenarios that could give rise to BUG.6, in which the box falls over or is pushed off the table while being loaded. The first scenario is that if the walls of the box tilt outward, or the bottom of the box is rounded, and the cargo is loaded at a point that is outside the region where the box is supported by the table, then the weight of the cargo may make the box tip over onto its side (Fig. 7.A). We can exclude this by requiring that the center of mass of every cargo object is above the convex hull of the contact points of the box with the table. It is easy to show that if this condition holds, then any tilting by the box will raise the height of the centers of mass of the cargo objects; therefore, the cargo objects cannot be exerting a force that causes the box to tilt.

A second scenario is illustrated in Fig. 7.B. By sliding down the slanted left-hand side of the box, $O1$ could exert sufficient leftward force on the box to push its end over the table. The center of mass of the pair $\{OB, O1\}$ would still be over the table, but if you now load a heavy object $O2$ on the left-hand side of the box, the box could fall off the table.

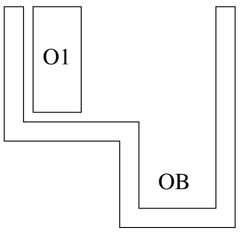
A third scenario is illustrated in Fig. 7.C: a cargo object $O1$ sliding or rolling around a curve exerts enough horizontal force to push the box over. (Note that if the side of the box is curved outward, then $O1$ exerts a centrifugal force whose magnitude is dependent on the velocity of $O1$.)

A fourth scenario is that, in the course of loading the box, the cargo objects settle and hit the side of the box in such a way as to knock the box over on its side. I find it hard to draw a convincing picture of this, so I leave this to the imagination of the reader.

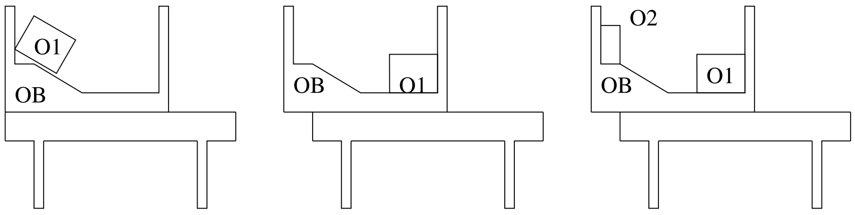
The first scenario is a quite plausible one, but as observed above, it is easy to find conditions that demonstrably exclude it. To exclude the remaining scenarios, we introduce the following default rule: If an object OB is stably supported on top of object OT and a collection of object OC is initially piled in heaps on top of OB, and the centers of mass of all the objects in OC remain over the convex hull of the points of contact between OT and OB, and OT remains motionless, then assume by default that OB remains motionless.

We have fallen back on a default rule here, rather than look for conditions that are demonstrably sufficient in a Newtonian theory for a number of reasons.

- The second, third, and fourth scenario are uncommon. Violations of this default rule will be very infrequent.
- It seems to be very difficult to find qualitative boundary constraints of the proper form — that is, constraints on shape and material properties, constraints that apply in the initial state, and constraints on the execution of the plan — that suffice to ensure that these scenarios are impossible, particularly since the third and fourth scenario have a complicated dependence on the velocities of the objects involved.

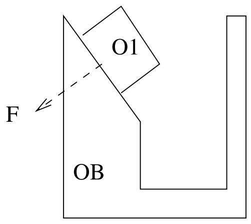


Falling over, case A



Time 1 Time 2 Time 3

Falling over, case B. Assume that all interactions are frictionless, and that O2 is much heavier than O1, which is much heavier than OB.



Falling over, case C

F is the force that O1 exerts on OB.

Fig. 7. Box falls over: three cases.

- If one were to compute such constraints they would almost certainly be far more restrictive than necessary. It seems pointless to impose highly restrictive conditions on the scope of our inference in order to exclude possibilities that are in any case very rare.

3.3.7. Why not use defaults for all our problems?

Since we are in any case resorting to default rules into order to deal with bugs 6, 7, and 8, why not do this for all the categories of bugs? In general, any place where we have required a condition to eliminate a bug, we could replace that, either with a default rule that the condition holds, or, even better, with a default rule that the bug doesn't arise. In some cases, this would clearly be unreasonable; it would be absurd to say that, by default, a given set of objects `uCargo` fits inside a given box `oBox`. On the other hand, it would be quite reasonable to replace the absolute condition excluding objects that can roll by a default rule that objects in general cannot roll. In the latter cases, whether or not to use a default rule would depend on what kinds of information are actually available and what kinds of situations actually arise in a given application. (There are, for instance, applications in which rolling objects are common.) Here, our objective has been to use as few different default rules as possible; namely, the two default rules discussed in the previous two sections, which were the only means we found to exclude bugs 6, 7.a, 7.c, 8.b, and 8.h. However we did not minimize the number of applications of this rule needed; as discussed in Section 3.3.4, we have used this default rule to exclude bugs 8.c through 8.f, whereas we could reasonably have excluded these monotonically by imposing further conditions.

3.4. Variants

We now turn to the variants on simple open boxes mentioned in the introduction: namely, boxes with lids, boxes with holes in the bottom and sides, and trays. First as regards lids, we observe that if you cover the box with a lid and you can guarantee that the lid will not come off during carrying, then you can be sure that none of the cargo items will come out of the box during carrying, thus eliminating BUG.7.c and all the subcategories of BUG.8. (That is the main point of having a lid.) You can also guarantee that no external object will enter the box while it is being carried, thus allowing the "isolation condition" COND.6 to be weakened. The plan, of course, must be modified to add the step of placing the lid onto the box between loading and carrying the box. We assume that the lid, like the objects in `uCargo`, is initially resting isolated on `oTable1` and that it fits on the box when the box is empty. However,

BUG.9 The lid may no longer fit on the box once the box is filled.
BUG.10 The lid may come off during carrying, either because
- a. It falls off on its own (consider, for example, a lid which is just a flat piece of cardboard laid over the top of the box).
- b. It is knocked off or removed by some external object during carrying.
- c. It is knocked off by the clattering of the cargo inside the box.

Boxes with holes suffer from the bug that

BUG.11 Objects may fall through these holes at any stage.

For trays and overfilled boxes, condition COND.3 is replaced by the condition

COND.3′ The cargo `uCargo` can be arranged as stable heaps supported by `oBox`.

and BUG.3 and BUG.4 are modified accordingly. However, we introduce a new subcategory of BUG.8:

BUG.8.i Depending on the stability of the heaps, even quite smooth motions or small tilts may cause a heap to collapse, potentially causing some object to fall off the heap.

The problem with trays is that it is very difficult to formulate reasonable qualitative constraints on object shapes that suffice to guarantee that the cargo can be piled in stable heaps on the tray; it is not even easy to guarantee that a single object stays put on a tray. (For example, if an object has a round bottom, then, even if it is weighted so that it can only roll very slightly and even if the tray is kept perfectly horizontal, the object can still gradually work its way off the tray in response to very small accelerations and decelerations in carrying the tray.) To formulate a reasonably general theory of trays, therefore, probably the right approach is to posit the qualitative condition that the cargo can be piled into stable heaps on the tray; in particular circumstances, this itself may be inferrable from fairly precise specifications of the geometry of the cargo. This indeed is the advantage of a box over a tray; once the cargo is inside the box, it does not matter whether the objects shift their positions or not, as long as they do not move violently enough to run into bugs 8.b though 8.h. Also, because of the walls of the box, stable positions of the cargo in the box can be made taller than on a tray and are easier to attain.

We fix the other bugs as follows:

BUG.9. We posit that the cargo fits inside the box, and that the lid does not extend into the inside of the box.
BUG.10. We posit that the lid caps the box; that it is heavy enough as compared to the impacts of objects inside and out that it is not knocked off; and that the trajectory during carrying is smooth enough that it is not flung off that way.
BUG.11. We posit that the objects do not fit through the holes.

Finally, if we consider the special case where the cargo consists of a single object, what is perhaps most striking is how many of the potential bugs still remain. Specifically, BUG.1, BUG.3.a, BUG.3.b, BUG.5, BUG,6, all the categories of BUG.8 except BUG.8.h, BUG.9, BUG.10, and BUG.11 are still problematic (though BUG.3.a can easily be fixed by positing that oBox does not "close in" on itself). BUG.2, BUG.3.c, BUG.4, BUG.7, and BUG.8.h no longer apply.

3.4.1. Bugs and their fixes: overall view

It may seem, on first glance, that once we have limited our theory by this large collection of qualifications and default rules there is not very much left. But in fact, most of these qualifications are essentially commonsensically obvious; a reasoner who sees that the cargo is too big, or that the carrying is very bumpy, or that the cargo can fall through holes in the bottom and so on will expect that the plan may well fail. Though the exact formulation is driven by the need to formalize and the desire to follow Newtonian physics as far as possible, these qualifications are, I would argue, basically part of a commonsense understanding of the domain.

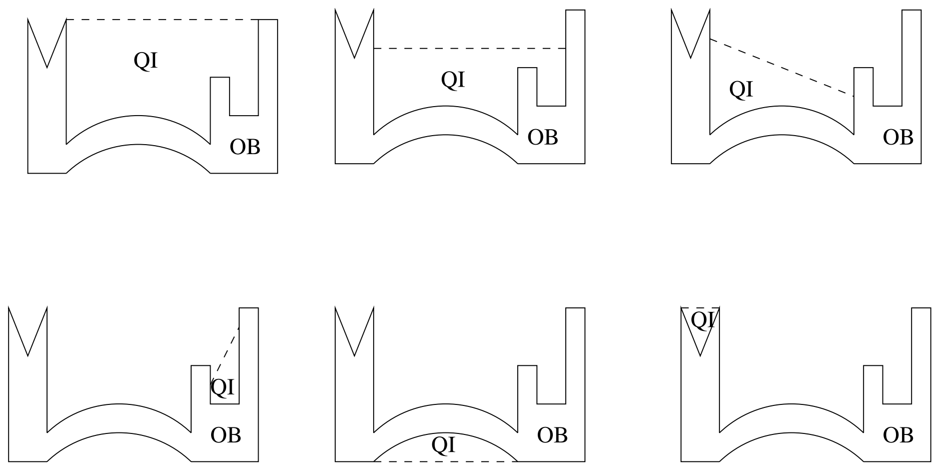


Fig. 8. OB can be construed as a box with in many different possible insides QI.

3.5. The geometry of a box

There are a few more issues that we want to address at the pre-formal level before setting forth to turn our theory into first-order logic. The first is the definition of what it means to be a box. We define an open box as a geometric predicate as follows:

Definition 3.5.1. A *region* is a set of points that is connected, bounded, and topologically regular.[11] The *topological boundary* of region R, denoted "Bd(R)", is the set of points in R that are not in the interior of R.

Definition 3.5.2. Let RB and RI be two regions. We say that RB is an *open box with inside* RI if the following two conditions hold:

- The interior of $RI \cap RB$ is empty. (RI is externally connected to RB [30].)
- Bd(RI)–Bd(RB), the part of the boundary of RI that is not part of the boundary of RB, lies in a plane and contains a circular disk of positive radius. (The latter condition is a *topological* condition to guarantee that the opening of the box is not a single point or a single curve but is a true face that a small enough object can get through.)

Note that a given shape RB can often be construed as an open box with many different possible values for RI (Fig. 8). Indeed, if RB forms an open box with one region of RI then necessarily there are many different possible values of RI, with faces running in many different directions, which are possible insides for RB. To give an intuition, if you can orient a solid object with shape RB in some position where it holds water, and then you fill some external cavity of OB with water to some depth, then the region occupied by the water constitutes a possible inside for RB.

Given a box RB with inside RI, the *outside* of $\langle RB, RI\rangle$ is the complement of $RB \cup RI$. An *opening* of $\langle RB, RI\rangle$ is a connected component of Bd(RI)–Bd(RB); all the openings of $\langle RB, RI\rangle$ lie in a single plane.

If $\langle RB, RI\rangle$ form a geometric box, solid object OB has shape RB, and solid object $O2$ is inside RI at one time and outside RI at a later time, then $O2$ must have exited through an opening of $\langle RB, RI\rangle$. Proof: Since $O2$ went from inside RI to outside RI it must have gone through the boundary of RI and it obviously can't go through the part the boundary of RI that borders OB; end of proof. Therefore, if $O2$ is initially inside RI, and the opening of $\langle RB, RI\rangle$ is always higher than $O2$, then it follows that $O2$ must remain inside RI.

3.6. Heaps and stability

A central concept in the qualitative physics of solid objects is that of a *stable heap* of objects. The precise definition of a "heap", a canonical example of a vague concept, is necessarily somewhat arbitrary, but the following serves our purposes here.

To begin with, in the commonsense setting we must distinguish between *mobile* objects, which can move, and *fixed* objects, which cannot. In a given state, a mobile object may be *grasped* by the agent; while it is being grasped, its motions are controlled by the agent and are not affected by external objects. Thus, neither fixed objects nor grasped objects are

[11] A region is topologically regular if it is equal to the closure of its interior. This essentially requires that the region has some "thickness" in three dimensions; it excludes curves, surfaces, shapes that are solid in some places and two-dimensional lamina in other places, and so on.

affected by the motions of mobile objects; their motions are boundary conditions for a prediction problem.[12] An object is *free-moving* in state S if it is mobile and not grasped.

A heap is a connected set of free moving objects; in our theory, unlike discussions of the Sorites paradox, a heap may contain a single object. A heap of objects is defined with respect to some set of supporting objects. Usually a heap actually has a single supporting object, such as the ground or a box, but in some cases a heap may rest on multiple supports, such as a board that is lying with ends on two different tables.

Definition 3.6.1. Let UH and US be disjoint sets of objects, where UH is non-empty. In state S, UH is a *heap* with supports US if the following conditions hold in S:

- All the objects in UH are free-moving.
- If $O1$ is in UH and object $O2$ abuts $O1$ then $O2$ is either in UH or in US.
- Each object in US abuts some object in UH.
- If OA and OB are in UH then there exists a sequence[13] $O_1 = OA$, $O_2 \dots O_k = OB$ such that $O_i \in UH$ and O_i abuts O_{i+1}.

A set UH is a *maximal heap* in state S if UH is a heap with supports US and all the objects in US are fixed or grasped. Equivalently, a maximal heap is a maximal collection of mobile, non-grasped objects connected by abutment; the supports of a maximal heap UH are all the fixed or grasped objects that abut some object in UH.

A number of consequences follow directly from this definition:

First, a fixed object or an object being grasped can only be a support; it cannot be part of a heap. A mobile object can be considered as part of a (non-maximal) heap or can be considered as a support; that is a matter of usefulness for the reasoner. For instance, if one is loading objects into a box, it is sometimes convenient to think of the objects inside as forming a heap (or several heaps) supported by the box, and it is sometimes convenient to think of the box together with the objects inside as forming a heap. Both viewpoints are OK.

Second, in any state, the maximal heaps constitute a partitioning of the free-moving objects into equivalence classes, where the equivalence relation is the transitive closure of the relation "OA and OB are both free-moving and abut".

Third, two distinct maximal heaps cannot abut one another. Thus, around any maximal heap there is a clear space free of any mobile objects. Since fixed objects and grasped objects are not affected by the motion of mobile objects, this means that a sufficiently small motion of any mobile object can only physically affect objects in the same heap. Similarly, any motion by a grasped OG can only affect mobile objects in heaps for which OG is a support. This gives us a very useful limit on the causal impact of events.

Finally, if UH is a set of mobile objects that are connected through abutment but are in free fall, then UH is a heap whose support is the null set of objects.

We next introduce the notion of a heap being *stably supported* in a given state. We will not give necessary and sufficient conditions for this, but we axiomatize a couple of the properties that we will need here:

Definition 3.6.2. Let $U1$ and $U2$ be sets of objects. and let H be a history. $U1$ is *isolated* from all objects except $U2$ if no object outside $U1 \cup U2$ comes into contact with any object in $U1$.

Axiom 3.6.1. If UH is stably supported by US in state S, then UH is a heap with supports US in S, and US is non-empty.

Axiom 3.6.2. Let H be a physically possible history with starting state $S1$. If heap UH is stably supported by US in $S1$, and US is motionless throughout H, and UH is isolated except for US and free moving, then UH is stably supported and motionless throughout H.

Axiom 3.6.3. Let H be a physically possible history that is temporally unbounded in the future (in Section 4.3 we will call this a "uhistory"). Let US be a set of objects that is motionless throughout H. Let UH be a set of mobile objects disjoint from US. Assume that the objects in UH are not grasped at any time in H. Then eventually either every object in UH is in a stable heap supported by some subset of US or some object in UH comes into contact with some object not in US. Equivalently, if UH is free moving and isolated in H except for US, then eventually every object in UH is in a stable heap supported by some subset of US.

(The above axioms are formalized as axioms H.1, H.2, and H.3 in Table 19.)

[12] There is an important and awkward exception to this. If the agent is "trying" to move a grasped object $O1$ in a given direction but that motion is blocked by a mobile object $O2$ that is "stuck", then moving $O2$ may permit $O1$ to proceed. A major difficulty in formulating an existence theorem for well-posed problems in the theory of rigid solid object dynamics with manipulators is to characterize under what circumstances and in what sense a specified manipulation constitutes a valid boundary condition.

[13] We assume throughout that there are only finitely many objects in the universe; hence all such sequences are finite.

This will be all we need for boxes. Reasoning about trays requires a more extensive theory of stability, which we will develop in future work.

3.7. No upward motion

Finally, we formulate the default rule that will allow us to infer that cargo objects do not come out of the top of the box. We want to state that by default objects in the box do not move upward with respect to the box; more generally, that objects in a heap by default do not move upward with respect to the supports of the heap. There can be different possible definitions of what is meant by object O "moving upward"; for convenience we will interpret this as meaning that the center of mass of O moves upward and we axiomatize the center of mass as some point within the convex hull of O. (If the region occupied by O is known, but the density distribution is entirely unknown, then all that can be said about O's center of mass is that it is in O's convex hull.)

Let us begin by formulating the rule in the special case where there is a single support object that maintains a constant vertical axis.

Preliminary Definition 3.7.1. In state $S1$, let O be an object in heap UH that is supported by a singleton object set $\{OS\}$. Let H be a history with starting state $S1$. Assume that OS maintains a constant vertical during H (that is, OS may translate and may rotate around the $\hat{z}$ axis, but it may not undergo a rotation that tilts the $\hat{z}$ axis; it may yaw but not pitch or roll.) Let QS be any point in OS and let QC be the center of mass of O. O *moves upward relative to* OS *in* H if height(QC)–height(QS) is larger at the end of H than at the beginning of H. (Since OS maintains a constant vertical, if this is true for any point QS in OS, it is true for all points in OS.) O has an *anomalous upward motion* in H if UH is free-moving and isolated from all objects except OS in H and O moves upward relative to OS in H.

A history is *upward motion free* if, for every subhistory $H1$ of H, no object has an anomalous upward motion in $H1$.

Default Rule 3.7.2. By default, any physically possible history is upward motion free.

We can now prove the following theorem:

Theorem 3.7.3. *In state S, let OB be a box with opening QT, let UH be a heap of objects supported by OB, and let O be an object in UH. Let F be the diameter of O, and let D be the vertical gap between QT and the center of mass of O. Let H be a physically possible history starting in S. Assume that throughout H, UH is free moving and isolated from all objects except OB, and that OB maintains a constant vertical, though it may be moved and rotated around the z-axis. Assume that $D > F$. Then by default O remain in OB throughout H.*

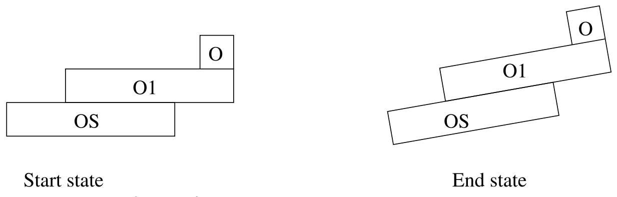
Sketch of proof. By default, H is upward motion free. Suppose that O goes out of the box during H. As argued in Section 3.5 above, if O goes out of the box, it must go through the opening at the top. Therefore, each point in O must be at the top at some point during H. Therefore at some time $T2$ during H, the center of mass of O must be closer to the top of OB it was at the start, so the center of mass of O has moved upward relative to OB in the subhistory of H that starts in S and ends at $T2$; but this contradicts the statement that H is upward motion free. □

To complete the proof, it is necessary to establish that the default conclusion is consistent, but it is easy to construct scenarios which are demonstrably consistent with the above givens, with Newtonian physics, and with all the axioms we shall state in Section 4 below, in which the objects in UH rest motionless at the bottom of OB throughout H. UH is a stable pile at the bottom of OB which demonstrably does not move relative to OB during all of H.

We next need to generalize the above rule in two respects. First, a heap may be supported by more than one support object. Second, the support object OS may rotate its vertical axis. In that case, even if the heap just sits quietly on top of the support, still the center of mass of some of the objects may move upward with respect to some of the points in OS. Indeed, if the center of mass of some object O in the heap does not lie above the convex hull of the horizontal projection of OS, then the center of mass of O may move upward with respect to *every* point in the support OS (Fig. 9).

As discussed above, in formulating our default rule, we try to keep it as restricted as possible, to lessen the likelihood of weird consequences. We therefore address the above two cases as follows. First, we consider an object in a heap as moving upward with respect to its supports only if it moves upward with respect to all its supports. Second, we deal with the problem of a support OS that rotates vertically as follows: For any history H, imagine taking a video of the behavior of the supports and the heaps during H. At any point during the video, you can take a vertical arrow and attach it to OS through any point in OS. Now replay the video from the beginning with that added arrow fixed to OS, so that it moves and rotates along with OS. We say that the heap object O *moves upward* with respect to OS in H if the projection of the center of mass of O onto the arrow is higher at the end of H than at the beginning of H.

Definition 3.7.4. In state $S1$, let O be an object in heap UH that is supported by an object set US Let H be a history with starting state $S1$. O "moves upward" with respect to US in H if the following condition holds: For every object $OS \in US$



$\{O, O1\}$ are a heap supported by OS.
Between the start and end state,
O moves upward with respect to every point in OS,
even though the whole group moves rigidly.

Fig. 9. Object in a heap moves upward with respect to every point in support.

and for every coordinate system QC "attached" to OS which is vertically aligned in some state in H, the z-coordinate in QC of the center of mass of O is larger at the end of H than at the beginning of H.

The definitions of "anomalous upward motion free", and of "upward motion free" in terms of "moves upward" are the same as in the Preliminary Definition 3.7.1, replacing "OS" by "US", and the Default Rule 3.7.2 remains unchanged. It can be seen that this deals reasonably, both with objects that stay fixed on the support, and with boxes. If an object stays in a constant position on the support, then, since any of these arrows are likewise fixed to the support, the projection of the center of mass on the arrow remains constant, consistent with the default rule.

In the scenario of the heap of objects in the box, we can weaken the statement that the box maintains a constant vertical to the constraint that the maximum deviation from the vertical is ϕ, if we correspondingly increase the required gap between the top of the heap and the opening of the box.

Theorem 3.7.5. (*This is the result cited in Section* 3.3.3.) *In state S, let OB be a box with opening QT, let UH be a heap of objects supported by OB, and let O be an object in UH. Let F be the diameter of O, and let D be the vertical gap between QT and the center of mass of O. Let G be the maximum distance between the projection onto the x–y plane of the center of mass of O the projection of any point in the opening of the box in S. Let H be a physically possible history starting in S. Assume that throughout H, UH is free moving and isolated from all objects except US, and that the vertical tilt of OB is never greater than ϕ. Assume that $D\cos(\phi) > G\sin(\phi) + F$. Then by default O remains in OB throughout H.*

Proof. It is easy to show geometrically that all of the opening is at least F higher than any of the boxes, where "height" is measured by projection on any of the tilted arrows that can be attached during H. The proof is otherwise identical to that of Theorem 3.7.3 above. This result is essentially Lemma 2.29 in our formal proof. □

On the other hand, if the box tips far enough over that objects that can spill out, then, relative to the arrow attached when the box is fully tipped, an object is moving downward, so the default is not violated.

We next discuss the workings of this default rule on a number of further examples. As we will see, the default rule does the "right thing" for Examples 1–3; in Example 4 it is not clear what the right thing should be but what the default rule does is at least not obviously wrong; in Example 5, the default rule is certainly not giving us what we would wish for. Throughout these examples, we will assume that the theory under discussion includes both the axioms we give below in Section 4 and also the axioms of the Newtonian physics of solid objects. Note that adding the axioms of Newtonian physics does not affect the inferences we have discussed above.

Example 1. Consider Fig. 10. Suppose that object OT is fixed, and objects OA and OB are mobile. What will actually happen here is that OA remains fixed and OB falls down. One might think that this would be a problem for the default rule, because by Definition 3.6.1, one can consider $\{OA\}$ to be a heap with supports $\{OT, OB\}$ and OA moves upward relative to OB. However, in fact the default is not a problem here, because the default rejects a behavior only if a heap object moves upward with respect to *all* its supports, and OA is not moving upward with respect to OT.

Example 2. Suppose that OA is in free fall inside box OB (Fig. 11). Then OB is not a support of any heap containing OA, so the default does not apply. That is not a problem. If it is known that OA is initially moving downward, then we can prove that it will eventually hit either OB itself or some other object that is (directly or indirectly) in contact with OB. At that point OA becomes part of a heap inside OB, so we can apply the default from that point. If it is known that OA is initially moving up, then it may well escape OB, if it is moving fast enough, so we do not want the default to apply. If the direction of motion of OA is not initially specified, there does not seem to be any very good reason to make the default assumption that it is not moving upward fast enough to escape, so the fact that the default rule does not apply is not a problem.

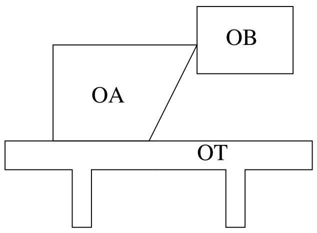


Fig. 10. Default rule: Example 1.

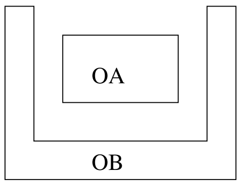


Fig. 11. Default rule: Example 2.

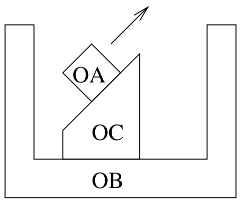


Fig. 12. Default rule: Example 3.

Example 3. Suppose that OB is a box, OC is a ramp resting at the bottom of OB, and OA is on OC sliding rapidly upward. Since $\{OA, OC\}$ is a heap with support $\{OB\}$, the condition of the defaults apply. However, if our theory includes the law of inertia, then OA must be moving upward with respect to the box, so the default is explicitly overridden, so again we get the correct answer (Fig. 12).

Similarly if one explicitly sets up and triggers a catapult, then the axioms of Newtonian physics imply that the missile will move upward, so the default rule is explicitly overridden.

Example 4. Suppose that object OA is inside OB but it is not specified whether it is part of a heap supported by OB or whether it is in free fall. Then the default rule applies, and gives rise to the conclusion that either OA is initially in a heap and remains in the box or it is initially in free fall and nothing can be said as to its motion. It is not clear to me whether this is the most desirable conclusion, or how one would decide what is the most desirable conclusion. Things get fairly nebulous at this level of ignorance. In any case, it seems to me that this is at least not indisputably wrong.

Example 5. Consider Fig. 13. The box is made of three jointed pieces: The first piece $OB1$ consists of the left-hand wall and the left-hand half of the floor. The second piece $OB2$ is the right-hand half of the floor, and the third piece $OB3$ is the right-hand wall. In the starting state, object OA rests on $OB1$ and $OB2$, so these are its supports. Now consider a history in which right-hand side of the box is folded upward and inward so that $OB2$ becomes vertical. Then the default rule applies but it does not prevent OA from exiting the box, for the following reason: If you attach the vertical arrow to $OB2$ at the end of the history, then that arrow is lying horizontally in the starting state, and therefore a motion of OA vertically upward would not be upward with respect to the arrow at that point. Therefore, such a motion would not be anomalous, and would be consistent with the statement that the history is upward motion free.

This example is unequivocally a failing of our theory; in this case, we definitely want to predict that OA does not exit the box, and our default rule is not strong enough to support that. This strongly suggests that we have not found the best possible formulation of the default rule, in the case of multiple supports with vertical rotations. It should be noted, though, that this is at least the right direction for failure; it is much better to have the default rule be too weak than too strong.

Finally, we need to specify a particular default logic in which the default rule can be stated and applied. It seems that Reiterian default theory [31] fits our needs better than circumscription [24]. Let me explain in terms of the standard example of birds that can fly. Suppose we have the following four first-order statements:

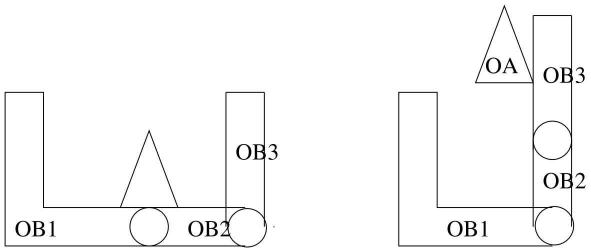


Fig. 13. Default rule: Example 5.

1. $\forall_X$ penguin(X) $\Rightarrow$ $\neg$canFly(X).
2. $\forall_X$ penguin(X) $\Rightarrow$ bird(X).
3. $\exists_X$ penguin(X).
4. bird(`tweety`).

We also have the default rule, "By default, birds can fly", and we would like to infer that Tweety can fly.

In the Reiterian theory, this default can be represented by the rule

5.R. bird(X) : canFly(X) / canFly(X).

The theory 1–4, 5.R supports the conclusion "canFly(`tweety`)" and therefore, from (1), $\neg$penguin(`tweety`).

But there is no way to get this out of circumscription. The default rule is represented by the first-order statement

5.C. bird(X) $\wedge$ $\neg$ab(X) $\Rightarrow$ canFly(X)

and the circumscriptive policy of minimizing the predicate "ab". But this does not support the conclusion canFly(`tweety`), whether or not the extension of "penguin" is allowed to vary, because a model where Tweety is the only penguin and the only abnormal entity does, in fact, minimize ab.

If you replace (3) above with "penguin(`fred`)", add the unique names assumption that `tweety`$\neq$`fred`, and adopt the circumscriptive policy of minimizing "ab" while allowing "penguin" to vary, then you can indeed deduce that Tweety can fly. But you can also deduce that Fred is the only penguin. This seems like a lot of bath water to hold onto for the sake of not losing the baby.

Going back to boxes: "canFly(X)" corresponds to "X comes out of the box", "bird(X)" corresponds to "X is in a heap on the floor of the box" and "penguin(X)" corresponds to "X is a missile being catapulted". The existential statement (3) corresponds to the fact that a complete theory of solid object dynamics should certainly support the inference that it is possible to set up and fire catapults. The impossibility of showing circumscriptively that Tweety is not a penguin corresponds to the impossibility of showing that the agent has not set up a catapult, given that he has loaded some objects into the box.

Therefore, it seems to me that, despite the attractiveness of circumscription — it is usually much easier to verify the correctness of proofs in circumscription than in Reiterian default theory — it is not applicable here. The same considerations apply to the default rule that we use to prove that the box will not fall over during loading.

4. The formal theory

We now proceed to encode the above theory in logical form. This is actually reasonably straightforward; except for the semantics of plans, we have covered all of the tricky issues already. Other than the two default rules H.5 and UP.1, defined below, we represent all of our theory in first-order logic. In this section we first establish a few basics: notational conventions and system of sorts (Section 4.1), and theories of real arithmetic and of Boolean operations on finite sets of objects (4.2). We will then describe the ontology, language, and theories that we will use for theories of time (4.3), space (4.4), motion (4.5), and physics (4.6). Section 4.7 presents a comprehension axiom for histories. Section 4.8 gives the semantics of our language of plans. Finally we give the specification of the problem to be solved (4.9) and the specification of the plan that solves it (4.10).

We assume the standard mathematical theories of integer and real arithmetic, Boolean operators on finite sets, Euclidean geometry, and continuous functions. Therefore, in our axiomatization, we will enumerate the sorts and symbols we need, but we will not list the axioms. In our formal proof, we cite theorems from these theories as needed without axiomatic proofs (though when the theorems are not obvious, we give proofs in the usual sense).

The axioms we enumerate are, for the most part, just those we need for the validation of the plan; we have not attempted anything like a complete axiomatization of these domains.

Table 2
Axioms of sorts.

SORT.1	The declaration of predicate symbol α as taking arguments of sorts $\sigma_1 \dots \sigma_k$ corresponds to the axiom $\forall_{X_1 \dots X_k} \alpha(X_1 \dots X_k) \Rightarrow \text{sortOf}(X_1, \sigma_1) \wedge \dots \wedge \text{sortOf}(X_k, \sigma_k)$. (Note: The equality symbol $X = Y$ has no sort declaration, so this axiom does not apply.)
SORT.2	The declaration of function symbol β as taking arguments of sorts $\sigma_1 \dots \sigma_k$ and returning a value of sort σ_0 corresponds to the two axioms $\forall_{X_0, X_1 \dots X_k} X_0 = \beta(X_1 \dots X_k) \neq \bot \Rightarrow$ $\text{sortOf}(X_1, \sigma_1) \wedge \dots \wedge \text{sortOf}(X_k, \sigma_k) \wedge \text{sortOf}(X_0, \sigma_0)$ and $\forall_{X_0, X_1 \dots X_k} \text{sortOf}(X_1, \sigma_1) \wedge \dots \wedge \text{sortOf}(X_k, \sigma_k) \Rightarrow \beta(X_1 \dots X_k) \neq \bot$.
SORT.3	The declaration of constant symbol γ as having sort σ corresponds to the axiom $\text{sortOf}(\gamma, \sigma)$.
SORT.4	$\forall_S \neg\text{sortOf}(\bot, S)$. (The null element has no sort.)
SORT.5	$\text{subsort}(S1, S2) \equiv \forall_X \text{ sortOf}(X, S1) \Rightarrow \text{sortOf}(X, S2)$. (Definition of subsort.)

4.1. Notational conventions

All this is pretty much standard and self-explanatory, but it is as well to lay it out systematically.

Our axioms are stated in a sorted first-order logic. We use symbols in Roman font beginning with a lower-case letter, such as "openBox", for function and predicate symbols, and symbols in typewriter font, such as `oBox`, for constant symbols. We use upper case symbols in italics for variables. Standard mathematical functions and predicates are used in the standard way; e.g. $X1 + X2$ is an infix function; the pair of curly brackets of $\{E1, E2\}$ is an outfix function (mapping $E1$ and $E2$ to the set $\{E1, E2\}$). To aid readability, the sort of a variable is indicated by the first letter; however, all axioms are stated so that this convention is not necessary. Quantified variables are subscripted to their quantifier. For brevity, we use two forms of restricted quantification in the subscript: a variable may be restricted by sort or by membership in a set. Logical operators follow the following precedence, from highest to lowest: negation $\neg$, conjunction $\wedge$, disjunction $\vee$, implication $\Rightarrow$, equivalence $\Leftrightarrow$, definitional equivalence $\equiv$, and quantifiers $\forall$ and $\exists$. Thus, the scope of a quantifier is to the end of the formula or to a right bracket that contains it. Free variables are taken to be universally quantified, where the scope of the implicit quantifier is the entire formula. Greek letters are infrequently used as meta-variables.

We have a hierarchical system of sorts, which is interpreted as syntactic sugar for a standard first-order theory with a null element $\bot$. (The null element is never used explicitly in our formulas.) In the translation to the base theory, an individual sort such as "object" or "state" is considered an entity. There are two predicates over sorts: "sortOf(X, S)", meaning that X is an entity of sort S, and "subsort($S1, S2$)", meaning that $S1$ is a subsort of $S2$. We will abuse notation in our formulas by writing "$S(X)$" where S is a sort instead of "sortOf(X, S)"; e.g. we will write "object(O)" instead of "sortOf(X,object)". The sorts of the arguments to predicate and function symbols and to the values of function and constant symbols will be declared in a self-explanatory notation when these symbols are defined. Table 2 shows the translation of sort declarations into the base language.

As axiom SORT.2 indicates, all functions are required to be total over the sort on which they are defined (except for numerical division, which is grandfathered). Hence, when we have a mapping which is single-valued but not total, we will use a relation symbol for it and add an axioms stating that it is single valued (when necessary). The "value" of the mapping will conventionally be the last argument. For this reason, in cases where we do use a function symbol, even if that is defined and not primitive, such as "startTime(J)" in Section 4.3 below, we do not have to add an axiom stating that the function is total over the sort; such an axiom is implicit from the use of the function symbol.

Axioms SORT.1–SORT.4 together imply that any ground term with argument $\bot$ evaluates to $\bot$ and that any ground atomic formula with argument $\bot$ is false, unless the predicate is the equals sign.

4.2. Real arithmetic; set theory

As stated above, we use real arithmetic and Boolean set theory over sets of elements. (As we shall see below, the sets that we need in this paper are sets of objects, and sets of geometric points; these two sorts are therefore defines as subsorts of "element".) Table 3 enumerates the sorts and the symbols used. All the symbols are entirely standard, except "count(U, I)", a predicate meaning that integer I is the number of elements in finite set U (if U is infinite, then this does not hold for any I). The letter after the sort is the one we will use to indicate variables of that sort.

4.3. Time

Our temporal theory in some respects resembles the phase-space theory of physics (and inherited by robotics from physics) rather than the situation-based theory more common in AI [32,27]. Specifically, for our purposes the state of the world at an instant can be characterized by the values of a fixed class of fluents. The history of the world is a function from real-valued time to states. A state may persist unchanged over a finite length of time, or a history may "return" to a previous state; whereas the model of situations is generally taken to be strictly forward-branching with no cycles. In our

Table 3
Sorts and symbols for real arithmetic and set theory.

Sorts:
Real numbers (X).
Integers (I). Subsort of reals.
Elements (E).
Sets of elements (U).
Symbols:
$X1 + X2$, $X1 - X2$, $X1 \cdot X2$, $X1/X2$, $\lfloor X \rfloor$, $\min(X1, X2)$, $X1 < X2$, $X1 \leqslant X2$, π.
$\emptyset$, $\{E1, E2 \ldots Ek\}$, $E \in U$, $U1 \cup U2$, $U1 \cap U2$, $U1 \subset U2$, $U1 - U2$.
count(U: set, I: integer).

model an action A is feasible in state S if there exists a history starting in S in which A occurs; whereas most models of situation calculus theories have explicitly forward-branching structures. Therefore, we will use the word "state" rather "situation" to denote a snapshot of the universe at an instant.

For brevity, we will treat clocktimes (e.g. May 1, 2006 12:45:00 PM) and durations (e.g. 1.57 hour) as real numbers, though of course a more rigorous treatment would take these to be separate sorts which have a real-valued measure in a given temporal coordinate system. For readability, we will use T for variables over clocktimes and D for variables over durations.

The other temporal sorts we will use are as follows:

- A *state* is a snapshot of the universe.
- A *fluent* is an entity that takes on different values in different states. A *Boolean fluent* such as "grasping(O)" (the agent is grasping object O) is true or false in a state; a *non-Boolean fluent* such as "place(O)" (the region occupied by object O) takes on values of some other sort. If a fluent Q takes on values of sort σ, we denote the sort of Q as "fluent[σ]"; e.g. grasping(O) has sort fluent[Bool] and place(O) has sort fluent[region]. In translating the sort language to the base language, the symbol "fluent" here is a function that maps a sort like "region" to the sort fluent[region].
- A *history* is a function from a closed interval of clocktimes to states.[14]
- Because we are using a language of plans that includes loops, which can be infinite loops, it is sometimes necessary to allow histories that are closed on the left but unbounded on the right. These are also convenient for stating that a certain condition is eventually attained. A *uhistory* is a function to states whose domain is an interval of clocktimes that is closed on the left and either closed or unbounded on the right. Thus, histories are a subsort of uhistories.

Tables 4–7 enumerate the temporal symbols, definitions, and axioms we use. The temporal theory actually requires only three primitives (in addition to the primitives on the reals): holds(S, Q), value(S, Q), and stateAt(J, T, S). However, it is useful to have a large vocabulary of defined predicates as convenient abbreviations.

It will also be convenient to define some additional syntactic conventions for constructing functions over fluents from functions and predicates over atemporal entities. First, if X is an atemporal entity, then we define value$(S, X) = X$ for all states S; that is, we conflate a fluent that are constant with its value.

Let $\Phi(X_1 : \sigma_1 \ldots X_k : \sigma_k)$ be a predicate (or equality sign) over atemporal sorts $\sigma_1 \ldots \sigma_k$. For $i = 1 \ldots k$ let Q_i be a fluent of sort fluent[σ_i]. Then we define $\Phi^{\#}(Q_1 \ldots Q_k)$ to be the Boolean fluent satisfying

$$\forall_{S:\text{state}} \text{holds}\big(S, \Phi^{\#}(Q_1 \ldots Q_k)\big) \Leftrightarrow \Phi\big(\text{value}(S, Q_1) \ldots \text{value}(S, Q_k)\big).$$

Similarly let $\Psi(X_1 \ldots X_k) : \sigma$ be a function with arguments of atemporal sorts $\sigma_1 \ldots \sigma_k$ and value of sort σ. Then $\Psi^{\#}(Q_1 \ldots Q_k)$ is the fluent of sort fluent[σ] satisfying

$$\forall_{S:\text{state}} \text{value}\big(S, \Psi^{\#}(Q_1 \ldots Q_k)\big) = \Psi\big(\text{value}(S, Q_1) \ldots \text{value}(S, Q_k)\big).$$

Finally, if Δ is a Boolean operator then $\Delta^{\#}$ is the corresponding function over Boolean fluents (used with the same syntax as the operator). That is,

$$\forall_{T:\text{time}} \text{holds}\big(T, Q1 \Delta^{\#} Q2\big) \Leftrightarrow \big[\text{holds}(T, Q1) \Delta \text{holds}(T, Q2)\big].$$

For example, if $Q1$ and $Q2$ are fluents whose value at each time is a set, then "$Q1 =^{\#} \emptyset$" is the Boolean fluent that holds at those times where $Q1$ is empty. "$Q1 \subset^{\#} Q2$" is the Boolean fluent that holds when $Q1$ is a subset of $Q2$. "$Q1 \cup^{\#} Q2$" is the fluent whose value at each time S is the union of the values of $Q1$ and $Q2$. "$Q1 \subset^{\#} Q2 \vee^{\#} Q2 \subset^{\#} Q1$" is the fluent that holds at all times in which either $Q1$ is a subset of $Q2$ or $Q2$ is a subset of $Q1$.

[14] Some of the axioms below, particularly DYN.12, HC.4, PLD.3, PLD.5, and PLD.7 could be stated more elegantly if we allowed the use of bounded, open intervals as well. It is not clear whether the theory would be simpler overall.

Table 4
Time: Sorts and symbols.

Sorts:
Time (T) = real.
Duration (D) = real.
State (S).
Fluent(Q).
History (H).
Uhistory (J).
Symbols:
holds(S: state, Q: fluent[Bool]). Boolean fluent Q holds in state S.
value(S: state, Q: fluent[γ]) $\rightarrow \gamma$. The value of non-Boolean fluent Q in state S. γ is a meta-variable ranging over sorts.
stateAt(J: uhistory, T: time, S: state). S is the state of J at time T.
timeIn(T: time, J: uhistory). T is a time in the domain of uhistory J.
startTime(J: uhistory) $\rightarrow$ time. The starting time of uhistory J.
endTime(H: history) $\rightarrow$ time. The ending time of history H.
start(J: uhistory) $\rightarrow$ state. The starting state of uhistory J.
end(H: history) $\rightarrow$ state. The ending state of history J.
unbounded(J: uhistory). Uhistory J is unbounded on the right (i.e. not a proper history).
stateOf(S: state, J: uhistory). S is a state attained by uhistory J.
throughout(J: uhistory, Q: fluent[Bool]). Boolean fluent Q holds throughout uhistory J.
throughoutxSE(J: uhistory, Q: fluent[Bool]). Q holds throughout uhistory J, except possibly at the start and end of J.
historySlice($J1$, $J2$: uhistory). $J1$ is a temporal slice of $J2$.
historyPrefix($J1$, $J2$: uhistory). $J1$ is a temporal prefix of $J2$.
historyProperPrefix($H1$:history,$J2$: uhistory). $H1$ is a proper prefix of $J2$.
historySuffix($J1$, $J2$: uhistory). $J1$ is a suffix of $J2$.
hsplice($H1$: history, $J2$, J: uhistory). J is the result of splicing $J2$ to the end of $H1$.
sameTime($J1$, $J2$: uhistory). $J1$ and $J2$ have the same clocktime interval as domain.
singleHist(S: state, H: history). H is a history consisting of a single instant at S.

Table 5
Temporal theory: Definitions.

Definitions:	These definitions are self-explanatory given the descriptions in Table 4.
TD.1	timeIn(T, J) $\equiv \exists_S$ stateAt(J, T, S).
TD.2	T = startTime(J) $\equiv$ timeIn(T, J) $\wedge$ [$\forall_{T1}$ timeIn($T1$, J) $\Rightarrow T \leqslant T1$].
TD.3	history(J) $\equiv$ $\exists_{T1}\forall_{T2}$ timeIn($T2$, J) $\Rightarrow T2 < T1$.
TD.4	T = endTime(H) $\equiv$ timeIn(T, H) $\wedge$ [$\forall_{T1}$ timeIn($T1$, J) $\Rightarrow T1 \leqslant T$].
TD.5	unbounded(J) $\equiv$ uhistory(J) $\wedge \neg$history(J).
TD.6	S = start(J) $\equiv$ stateAt(J,startTime(J),S).
TD.7	S = end(H) $\equiv$ history(H) $\wedge$ stateAt(H,endTime(H),S).
TD.8	stateOf(S, J) $\equiv \exists_T$ stateAt(J, T, S).
TD.10	throughout(J, Q) $\equiv \forall_S$ stateOf(S, J) $\Rightarrow$ holds(S, Q).

4.4. Space

The ontology we use for space is Euclidean geometry ($\Re^3$). The spatial language is constructed entirely *ad hoc*. That is, Table 8 enumerates the sorts and the predicates that we use in the physical axioms of Section 4.6 and in the problem statement of Section 4.9; it does not attempt any systematic discussion of geometric reasoning. We give here the formal definition of the predicates "openBox", "partlyAbove" and "altogetherAbove" but otherwise do not list any geometric axioms or definitions, which are all standard. In our formal proof, we will cite standard or easily proved geometric theorems as needed.

The large number of geometric sorts that we use here may startle readers who are used to more ontologically pure theories such as [3], in which the only geometric sort allowed is the sort of well-behaved, fully dimensional regions. However, it turns out that, strictly speaking, the greater ontological profligacy here is illusory. As I have shown in [9], if you have a first-order language that allows quantification over regions, then even if you restrict the language to the single predicate "closer(X, Y, Z)", meaning "region X is closer to Y than to Z" and you restrict the universe of entities to include only simple polygons, nonetheless all of these ontological categories and the whole range of standard geometric concepts can be defined as first-order constructions in this language.

Table 6
Temporal theory: More definitions.

TD.11	$\text{throughoutxSE}(J, Q) \equiv \forall_T\ \text{timeIn}(T, J) \wedge \text{stateAt}(J, T, S) \wedge \text{startTime}(J) < T \wedge [\text{unbounded}(J) \vee T < \text{endTime}(J)] \Rightarrow \text{holds}(S, Q).$
TD.12	$\text{historySlice}(JA, JB) \equiv \forall_T\ \text{timeIn}(T, JA) \Rightarrow [\text{timeIn}(T, JB) \wedge \forall_S\ \text{stateAt}(JA, T, S) \Leftrightarrow \text{stateAt}(JB, T, S)].$
TD.13	$\text{hSlice}(J, T1, T2, H) \equiv \text{historySlice}(H, T) \wedge \text{startTime}(H) = T1 \wedge \text{endTime}(H) = T2.$
TD.14	$\text{historyPrefix}(JA, JB) \equiv \text{historySlice}(JA, JB) \wedge \text{startTime}(JA) = \text{startTime}(JB).$
TD.15	$\text{historyProperPrefix}(HA, JB) \equiv \text{historyPrefix}(HA, JB) \wedge [\neg\text{unbounded}(JB) \Rightarrow \text{endTime}(HA) < \text{endTime}(JB)].$
TD.16	$\text{historySuffix}(JA, JB) \equiv \text{historySlice}(JA, JB) \wedge [[\text{unbounded}(JA) \wedge \text{unbounded}(JB)] \vee \text{endTime}(JA) = \text{endTime}(JB)].$
TD.17	$\text{hsplice}(H1, J2, J) \Leftrightarrow \text{historySlice}(H1, J) \wedge \text{historySlice}(J2, J) \wedge \text{startTime}(J2) = \text{endTime}(H1).$
TD.18	$\text{sameTime}(J1, J2) \equiv \forall_T\ [\text{timeIn}(T, J1) \Leftrightarrow \text{timeIn}(T, J2)].$
TD.19	$\text{singleHist}(H, S) \equiv \text{startTime}(H) = \text{endTime}(H) \wedge S = \text{start}(H).$

Table 7
Temporal theory: Proper axioms.

T.1	$\text{timeIn}(T, J) \Rightarrow \exists^1_S\ \text{stateAt}(J, T, S).$ (A history J has only one state S at a given time T.)
T.2	$\forall_{T,J}\ \text{timeIn}(T, J) \Leftrightarrow \text{startTime}(J) \leqslant T \wedge [\text{unbounded}(J) \vee T \leqslant \text{endTime}(J)].$ (A history J has a state for every time T between its start time and end time. An unbounded uhistory has a state for every time after its start time.)
T.3	$\forall_{S,T}\ \exists_H\ \text{singleHist}(H, S) \wedge \text{startTime}(H) = T.$ (One can construct an instantaneous history corresponding to any state S.)
T.4	$\forall_{J,T1,T2}\ \text{timeIn}(T1, J) \wedge \text{timeIn}(T2, J) \wedge T1 \leqslant T2 \Rightarrow \exists_H\ \text{historySlice}(H, J) \wedge \text{startTime}(H) = T1 \wedge \text{endTime}(H) = T2.$ (One can slice history J at any times $T1, T2$ within the scope of J).
T.5	$\forall_{H1,H2}\ \text{end}(H1) = \text{start}(H2) \wedge \text{endTime}(H1) = \text{startTime}(H2) \Rightarrow \exists_H\ \text{hsplice}(H1, H2, H).$ (Any two histories that meet properly can be spliced together.)

Because of the large number of geometric sorts, we are less systematic about the use of initial letters to indicate sort of variables. In most cases, variables of sort "region" start with R; points start with P; pointSets start with PS; rigid mappings start with M; distances start with D (the ambiguity with durations should not cause problems); coordinate systems start with C; other geometric variables start with G.

A *coordinate system* is a standard three-dimensional orthogonal right-handed coordinate system. We assume a fixed unit of length for all coordinate systems. Different coordinate systems may differ in orientation and in the choice of origin.

A *region* is a spatial region that could be the shape of a physical object. We posit that a region is topologically regular (i.e. equal to the closure of its interior), bounded, and connected.

A *rigid mapping* is a positive, orthonormal mapping of three-dimensional space to itself; that is, the composition of a rotation and a translation. (Reflections are not allowed.)

The other geometric sorts are self-explanatory.

We use some of the RCC [30] topological relations between regions. However, since our vocabulary of symbols is so large, we preface the name with "rcc". Thus, the predicate "rccC($R1, R2$)" is the relation usually designated "C($R1, R2$)" in the qualitative spatial reasoning literature; namely, regions $R1$ and $R2$ are connected. Similarly "rccEC($R1, R2$)" is the QSR relation EC($R1, R2$), $R1$ and $R2$ are externally connected; "rccDC($R1, R2$)" is the QSR relation DC($R1, R2$), $R1$ and $R2$ are disconnected; "rccDR($R1, R2$)" is the QSR relation DR($R1, R2$), $R1$ and $R2$ are disjoint (either disconnected or externally connected); and "rccO($R1, R2$)" is the QSR relation O($R1, R2$), $R1$ and $R2$ overlap. The RCC relations are only applied to regions, not to other point sets.

The function "mappingImage(M, G)" denotes the image of G under rigid mapping M. The sort of mappingImage(M, G) is the same as the sort of G (in other words, all our geometric sorts are closed under rigid mappings).

The predicate "diameter(PS, X)" means that X is the diameter of point set PS; that is, the least upper bound on distance($P1, P2$) where $P1$, $P2$ are points in PS. If PS is unbounded, then this does not hold for any X.

The predicate "altogetherAbove($PS1, PS2$)" means that every point in $PS1$ is above some point in $PS2$; and no point in $PS2$ is above any point in $PS1$ (Table 9).

The predicate "partlyAbove($PS1, PS2$)" means that some point in $PS1$ is above some point in $PS1$.

Table 8
Spatial sorts and symbols.

Sorts:
distance (D) = real.
angle = real.
geomEntity (G).
Any geometric entity. This is a supersort of all the sorts enumerated below.
point (P). Subsort of element.
pointSet (PS). Any set of points. Subsort of set.
vector.
coordinateSystem.
region (R). Subsort of pointSet.
rigidMapping (M).

Symbols:
rccC($R1, R2$: region).
rccEC($R1, R2$: region).
rccDC($R1, R2$: region).
rccDR($R1, R2$: region).
rccO($R1, R2$: region).
mappingImage(M: rigidMapping, G: geomEntity) $\rightarrow$ geomEntity.
The image of G under M.
boundary(R: region) $\rightarrow$ pointSet. The boundary surface of R (R-interior(R)).
planar(PS: pointSet). PS lies in a plane.
openBox(RB, RI: region, PS: pointSet).
RB is a box with inside RI and opening PS (Section 3.5).
diameter(R: region) $\rightarrow$ distance.
$\hat{z}$: vector. The absolute upward direction.
zAxis(GC: coordinateSystem) $\rightarrow$ vector. The z axis of coordinate system GC.
zCoor(P: point, GC: coordinateSystem) $\rightarrow$ real. The z-coordinate of P in GC.
disk(PS: pointSet). PS is a two-dimensional solid disk.
convexHull(PS: pointSet) $\rightarrow$ pointSet.
pointAbove($P1, P2$: point). $P1$ is vertically above $P2$.
partlyAbove($PS1, PS2$: pointSet).
altogetherAbove($PS1, PS2$: pointSet).
height(P: point) $\rightarrow$ distance.
top(R: region) $\rightarrow$ distance. D is the maximum value of height(P) for $P \in R$.
bottom(R: region) $\rightarrow$ distance. D is the minimum value of height(P) for $P \in R$.
xyProj(PS: pointSet) $\rightarrow$ pointSet.
cuboid(R: region, L, D, H: distance).
verticalTilt($M1, M2$: rigidMapping) $\rightarrow$ real.
cos(R: real) $\rightarrow$ real.
sin(R: real) $\rightarrow$ real.

Table 9
A few spatial definitions.

Definitions:
SD.1 openBox(RB, RI, PS) $\equiv$
rccEC(RB, RI) $\wedge$ $PS =$ boundary(RI) $-$ boundary(RB) $\wedge$ planar(PS)$\wedge$
$\exists_{PSD}\ PSD \subset PS$ $\wedge$ disk(PSD).
SD.2 partlyAbove($PS1, PS2$) $\equiv$
$\exists_{P1,P2}$ pointIn($P1, PS1$) $\wedge$ pointIn($P2, PS2$) $\wedge$ pointAbove($P1, P2$).
SD.3 altogetherAbove($PS1, PS2$) $\equiv$
$\neg$partlyAbove($PS2, PS1$) $\wedge$ $\forall_{P1 \in PS1}\ \exists_{P2 \in PS2}$ pointAbove($P1, P2$).

It is useful to posit a standard coordinate system, with a vertical z-axis. The function "height(P)" is the height of point P in the standard coordinate system. The function "xyProj(PS)" is the projection of point set PS in the standard coordinate system. The predicate "cuboid(R, L, D, H)" means that region R is a cuboid with length L, depth D, and height H (L and D need not be aligned with the standard coordinate axes).

The meanings of the remaining symbols in Table 8 are obvious. Functions and predicates defined over point sets are overloaded to apply to individual points by coercing the point P to the point set $\{P\}$.

The function "verticalTilt(M_1, M_2)" (used in axiom P1.17, Section 4.10) is the angle between the vertical direction $\hat{z}$ and the direction $M_2(M_1^{-1}(\hat{z}))$. The significance is as follows: Suppose that QV is a pseudo-object vector with source O. Let $M1 =$ value($S1$, placement(O)) and let $M2 =$ value($S2$, placement(O)). If QV points upward in $S1$, then verticalTilt($M1, M2$) is the co-latitude of QV in $S2$ (that is, the angle between the direction of QV in $S2$ and the vertical axis).

Table 10
Kinematics: Symbols.

Sorts:
object (O).
pseudo.
gObject (Q). Supersort of object and pseudo.
objectSet (U). Set of objects.
Symbols:
source(O: gObject) $\rightarrow$ object.
shape(Q: gObject) $\rightarrow$ geomEntity.
placement(Q: gObject) $\rightarrow$ fluent[rigidMapping].
place(Q: gObject) $\rightarrow$ fluent[geomEntity].
motionless(J: uhistory, O: object).
fixed(O: object).
mobile(O: object).
objectsOf(S: state) $\rightarrow$ objectSet.
objectsOf(J: uhistory) $\rightarrow$ objectSet.
(For readability, we overload the function “objectsOf”.)
`kinematicState` $\rightarrow$ fluent[Bool].
kinematic(J: uhistory).
empty(R: region) $\rightarrow$ fluent[Bool].
mappingDistance($M1$, $M2$: rigidMapping; P: point).

Table 11
Kinematics: Definitions.

Definitions:	
KD.1	value(S,place(Q)) = mappingImage(value(S,placement(Q)),shape(Q)).
KD.2	motionless(J, O) $\equiv$ $\forall_S$ stateOf(S, J) $\Rightarrow$ value($S1$,placement(O)) = value(start(J),placement(O)).
KD.3	mobile(O) $\equiv \neg$fixed(O).
KD.4	holds(S,`kinematicState`) $\Leftrightarrow$ $\forall_{O1,O2 \in \text{objectsOf}(S)}$ holds(S,rccDR$^{\#}$($\uparrow O1, \uparrow O2$)).
KD.5	kinematic(J) $\equiv$ throughout(J,`kinematicState`) $\wedge$ $\forall_{O \in \text{objectsOf}(J)}$ fixed(O) $\Rightarrow$ motionless(H, O).
KD.6	holds(S,empty(R)) $\Leftrightarrow$ $\forall_{O \in \text{objectsOf}(S)}$ holds(S,rccDR$^{\#}$($\uparrow O$, R)).
KD.7	mappingDistance($M1$, $M2$, P) = distance(mappingImage($M1$, P),mappingImage($M2$, P)).

4.5. Kinematic theory of objects in motion

We can now formulate the kinematic theory of rigid objects in motion (Tables 10–12). We introduce three new sorts. An *object* is a rigid solid object. We assume that all objects are disjoint; we do not allow one object to be part of another. A *pseudo-object* [6] is a geometric entity that “moves around” with an object, such as the center of mass of an object, the hole of a donut, the apex of a cone, and so on. The *source* of pseudo-object Q is the object to which Q is “attached”. A *generalized object* (gObject) is a supersort that includes both objects and pseudo-objects.

We characterize an object O and its associated pseudo-objects in terms of an arbitrary standard position. The *shape* of O is the region that it occupies in its standard position. The shape of pseudo-object Q is the geometric entity that instantiates Q when O is in its standard position.

Any two possible positions of a rigid object O are related by a rigid mapping; that is, a combination of a translation and a rotation. For any object O and state S, we define the *placement of O in S*, denoted value(S,placement(O)), as the rigid mapping from the standard position to the position in S. The sort of placement(O) is thus fluent[rigidMapping]. The region occupied by O in state S is the image of shape(O) under the mapping value(S,placement(O)); this is denoted value(S,place(O)) (definition KD.1). The same holds for pseudo-objects; for any state S and pseudo-object Q, the place of Q in S is the image under a rigid mapping of the shape(Q). By constraining this rigid mapping to be the placement in S of the source of Q (axiom K.2), we enforce the condition that the pseudo-object “moves together” with the associated object.

Since the function place(O) is used so frequently in our theory, we abbreviate it using the symbol $\uparrow O$. For example, the formula, “holds(S,rccEC$^{\#}$($\uparrow O1, \uparrow O2$))” is an abbreviation for “holds(S,rccEC$^{\#}$(place($O1$), place($O2$)))”, meaning that $O1$ and $O2$ are externally connected in state S. Also, by convention, we extend any geometric predicate Φ to objects and pseudo-objects by defining $\Phi(Q_1 \ldots Q_k) = \Phi(\text{shape}(Q_1) \ldots \text{shape}(Q_k))$. For instance, if O is an object, then “cuboid(O)” is equivalent to “cuboid(shape(O))”.

A state S is *kinematic* if no two objects occupy overlapping regions in S. A history H is kinematic if all states in H are kinematic and all fixed objects are motionless. We do not posit that all histories are kinematic, because for some purposes it is useful to contemplate hypothetical histories that are not kinematic. For instance, the easiest way to define an impact between two objects is to assert that, if the objects continued with the same velocities, they would interpenetrate; and the easiest way to express that contrary-to-fact conditional is in terms of a hypothetical, non-kinematic history in which the

Table 12
Kinematic theory: Axioms.

Axioms:

K.1 object(O) $\Rightarrow$ source(O) $= O$.
K.2 $\forall_Q$ placement(Q) = placement(source(Q)).
K.3 object(O) $\Rightarrow$ region(shape(O)).
K.4 stateAt(J, T, S) $\Rightarrow$ objectsOf(J) = objectsOf(S).
K.5 $\forall_{DE:\text{distance}, P:\text{point}, J, O, T1, S1}$
$O \in$ objectsOf(J)$\wedge$ stateAt($J, T1, S1$) $\wedge\ 0 < DE \Rightarrow$
$\exists_{DD:\text{duration}}\ 0 < DD\ \wedge$
$\forall_{T2,S2}\ T1 - DD < T2 < T1 + DD\ \wedge$ stateAt($J, T2, S2$) $\Rightarrow$
mappingDistance(value($S1$,placement(O)),
value($S2$,placement(O)),P)
$< DE$.
(Every object O moves continuously in every history J. This is the usual $\epsilon - \delta$ definition of continuity. DE is ϵ, DD is δ.)

Table 13
Axioms of grasping.

Symbols:
freeGrasp $\rightarrow$ fluent[Bool].
grasping(O: object) $\rightarrow$ fluent[Bool].

Definition:
GD.1 holds(S,freeGrasp) $\Leftrightarrow \neg\exists_O$ holds(S,grasping(O)).

Axioms:
G.1 holds(S,grasping($O1$)) $\wedge$ holds(S,grasping($O2$)) $\Rightarrow O1 = O2$.
(The agent grasps at most one object at a time.)
G.2 holds(S, grasping(O)) $\Rightarrow O \in$ objectsOf(S).

two objects do continue with the same velocity and do interpenetrate. (In Lemma 2.13 of our formal proof, we actually do use such hypothetical histories to simplify the proof.)

We do posit (axiom K.5 below) that objects move continuously in every history; we have not found any use for discontinuous histories.

4.6. Physical theory

We now turn to the physical theory, which is new in this paper. (The previous theories are not.) Here we axiomatize some of the properties of *dynamic* histories; that is, histories that obey the laws of the dynamics of solid objects. However, unlike kinematic histories, we do not give necessary and sufficient conditions for a history to be dynamic.

We divide this section into four parts: The theory of grasping (Section 4.6.1), general characteristics of dynamic histories (Section 4.6.2), the theory of stable heaps (Section 4.6.3), and the default rule prohibiting anomalous upward motion (Section 4.6.4).

4.6.1. Grasping

As described in Section 3.1, we use a very rudimentary theory of grasping in this paper (Table 13). The agent can grasp one object at a time, or may not be grasping anything.

It will be convenient, for technical reasons, to assume that any state of grasping takes place over an open time interval. This can be related to a more realistic theory of manipulators if we define "grasping" to mean that the manipulators are exerting a positive force on the object; if the force is a continuous function of time, then it will be greater than zero over an open time interval.[15] However, though this definition works well in straightforward cases, it may break down in more complicated cases; for instance if the agent tries to ungrasp an object when it is not otherwise stably supported. In such cases, the idealization that there is a simple Boolean fluent "grasping" does not apply.

4.6.2. Dynamic histories

In this section we describe some general axioms that govern dynamic histories. We introduce the predicate "dynamic(J)" meaning that history J obeys the laws of the dynamic theories of rigid solid objects (Tables 14, 15). We do not attempt to characterize necessary and sufficient conditions for this, but merely state those axioms that we will use in our example here.

[15] Extending this idea, an alternative approach to axiomatizing grasping would be to posit a continuous real-valued fluent "graspForce(O)", the grasping force that the agent exerts on object O, and to define "grasping(O)" as holding in states where graspForce(O) > 0. This would simplify the theory in some respects and complicate it in others; it is not clear whether overall it would be a gain.

Table 14
Dynamics: Symbols.

Symbols:
dynamic(J: uhistory).
isolated(UH, US: objectSet) $\rightarrow$ fluent[Bool].
sameStateOn($S1$, $S2$: state, U: objectSet).
sameStateExcept($S1$, $S2$: state, U: objectSet).
sameHistoryOn($J1$, $J2$: uhistory, U: objectSet).
sameMotionOn($J1$, $J2$: uhistory, U: objectSet, D: duration).
sameUntilEnd($H1$, $H2$: history).
coherentGrasping($J1$, $J2$: uhistory). See definition DYD.7 and axiom DYN.9.
freeMotion(O: object) $\rightarrow$ fluent[Bool].
movingThroughout(O: object, H: history).
parallelMotion($O1$, $O2$: object, H: history).

Following our discussion in Section 3.6, we define the fluent "isolated(UH, US)". In state S, a collection of objects UH is isolated from all objects except US if no object in UH is in contact with any object outside of $UH \cup US$. This fluent enables us to posit that a collection of objects is free from interference from other objects without needing to impose draconian closed-world assumptions that demand that such other objects do not exist. This boundary condition is important both in the verification of our plan and in the statement of physical axioms.

Another useful concept is that of two states $S1$ and $S2$ being the same on the set of objects U; in our theory, this holds if the positions of every object in U is the same in $S1$ as in $S2$ and object O in U is being grasped in $S1$ if and only if it is being grasped in $S2$. The concepts of two states being the same except on the objects in U, or of two histories being the same on the objects in U are defined analogously.

Axioms DYN.1 and DYN.2 of Table 16 both state that new states or histories can be constructed from old ones by various kinds of modifications. Axioms DYN.3–DYN.8 discuss how a dynamic histories can be constructed and modified. Note that not all the modification operators that apply to histories in general, or even to kinematic histories, apply to dynamic histories. For instance the projection of a history H onto a subset of its objects is a history $H1$ (axiom DYN.2) and if H is kinematic then trivially $H1$ is kinematic, but it is not the case that if H is dynamic then $H1$ is dynamic. For instance, H may have mobile object `o1` supported on fixed object `o2`; if you project onto just `o1`, then `o1` is floating in mid-air. However, DYN.7 gives sufficient conditions under which the projection of a dynamic history is itself dynamic: if the objects US are all fixed or grasped throughout H, and the objects UM are isolated except for US, then the projection of H onto the object set $UM \cup US$ is dynamic.

DYN.8 is a converse, of a sort, to DYN.7. It states that if $J1$ and $J2$ are dynamic histories which are consistent in the sense that no objects in $J1$ overlaps any object in $J2$ at any time, that any objects in common between the two are placed in the same place, and if the agent is grasping $O1$ in $J1$, then he is not at the same time grasping $O2$ in $J2$, then one can "play" $J1$ and $J2$ "side by side" and the combined history is itself dynamic. One might wonder whether the constraint that no two objects overlap is not too weak; should we not have to require that no two objects from $J1$ and $J2$ come into contact? After all, two solid objects do not have to collide to affect one another's behavior, they merely have to come into contact. The answer is this: Suppose that $O1$ follows a trajectory in $J1$ that comes into contact but does not overlap the trajectory of object $O2$ in $J2$. Then the forces on $O1$ from the objects in $J1$ are sufficient to account for its trajectory, and likewise for $O2$, so there need not be any normal force between $O1$ and $O2$ (except possibly in the end state of J, but that has no consequences in J). If there is no normal force, then there is no frictive force either.

Axioms DYN.10–DYN.13 (Table 17) characterize the agent's ability to choose to grasp and ungrasp. In this representation, the feasibility of an action is expressed as the existence of a dynamic history in which the action is carried out. This is analogous to branching models of time, such as the situation calculus, in which the feasibility of action A in situation S is expressed as the existence of a successor state $S1$ such that A transforms S into $S1$. In dealing with continuous time, there are generally two cases to be considered; first, the continuance or beginning of an action *at* a given time, following some previous history; and, second, the continuance or beginning of an action *immediately after* a given state. Axioms DYN.10–DYN.13 describe these two options for grasping and ungrasping. Axiom DYN.10 states that at the end of any dynamic history, the agent can always choose to free his grasp. This is expressed by the rule that if H is a dynamic history, then there exists another dynamic history $H1$ which is identical to H except that, in end($H1$), the agent releases whatever he was holding in end(H). Axiom DYN.11 states that if the agent has been grasping O throughout H up until the end, he can continue to grasp O at the end. Axiom DYN.12 states that if the agent's grasp is free in S then he can continue not to grasp anything throughout some history starting in H. Axiom DYN.13 states that if the agent's grasp is free in S or if he is grasping O in S, then he can grasp O in some history starting immediately after S.

In a similar way, DYN.14 gives a sufficient condition for the feasibility of manipulating an object O along a given trajectory HK: If history HK is kinematically possible, and all the objects in HK either move parallel to object O or are motionless, then at least an initial segment of the motion of O in HK is dynamically possible; that is, there exists a dynamically possible history $H2$ starting in start(HK) in which O follows the same motion as in HK. In the box example, suppose that S is the state when all the cargo is loaded, and HK is a trajectory of carrying the box along a specified path. Then HK would be the history in which the objects inside the box keep a fixed position relative to the box and all other objects

Table 15
Dynamics: Definitions.

Definitions:

DYD.1 holds(S,isolated(UH, US)) $\Leftrightarrow$
$UH \cap US = \emptyset \wedge$
$\forall_{OH \in UH, OS \in \text{objectsOf}(S)}$ holds(S, rccC$^{\#}$($\uparrow OH, \uparrow OS$)) $\Rightarrow OS \in US \cup UH$.

DYD.2 sameStateOn($S1, S2, U$) $\equiv$
$U \subset$objectsOf($S1$) $\wedge$ $U \subset$objectsOf($S2$) $\wedge$
[$\forall_{O \in U}$ value($S1$,placement(O)) = value($S2$,placement(O)) $\wedge$
[holds($S1$,grasping(O)) $\Leftrightarrow$ holds($S2$,grasping(O))]].

DYD.3 sameStateExcept($S1, S2, U$) $\equiv$
objectsOf($S1$)$-U$ = objectsOf($S2$)$-U \wedge$ sameStateOn($S1, S2$,objectsOf($S1$)$-U$).

DYD.4 sameHistoryOn($J1, J2, U$) $\equiv$
sameTime($J1, J2$) $\wedge$
[$\forall_{T,S1,S2}$ stateAt($J1, T, S1$) $\wedge$ stateAt($J2, T, S2$) $\Rightarrow$ sameStateOn($S1, S2, U$)].

DYD.5 sameMotionOn($J1, J2, U, D$) $\equiv$
$U \subset$objectsOf($J1$) $\cap$ objectsOf($J2$) $\wedge$
$\forall_{T,S1,S2,O}$ stateAt($J1, T+D, S1$) $\wedge$ stateAt($J2, T, S2$) $\wedge$ $O \in U \Rightarrow$
value($S1$,placement(O)) = value($S2$,placement(O)).
(The objects in U have the same motion in $J2$ as in $J1$ with time shift D, though not necessarily the same grasping relations.)

DYD.6 sameUntilEnd($H1, H2$) $\equiv$
sameTime($H1, H2$) $\wedge$
$\forall_{T,S} T <$ endTime($H1$) $\wedge$ stateAt($H1, T, S$) $\Rightarrow$ stateAt($H2, T, S$).

DYD.7 coherentGrasping($J1, J2$) $\equiv$
$\forall_{O1,O2,T,S1,S2}$ stateAt($J1, T, S1$) $\wedge$ stateAt($J1, T, S2$) $\wedge$
holds($S1$,grasping($O1$)) $\wedge$ holding($S2$,grasping($O2$)) $\Rightarrow O1 = O2$.
($J1$ and $J2$ do not place conflicting conditions on what the agent is grasping at a given time. Condition of axiom DYN.9.)

DYD.8 holds(S,freeMotion(O)) $\Leftrightarrow$
$O \in$objectsOf(S) $\wedge$ mobile(O) $\wedge$ $\neg$holds(S,grasping(O)).

DYD.9 parallelMotion($O1, O2, H$) $\equiv$
$\forall_S$ stateOf(S, H) $\Rightarrow$
$\exists_{M:\text{rigidMapping}}$ value(S,placement($O1$)) =
mappingImage(M,value(start(H),placement($O1$))) $\wedge$
value(S,placement($O2$)) =
mappingImage(M,value(start(H),placement($O2$))).

Table 16
Basic properties of dynamics: Axioms (beginning).

Axioms:

DYN.1 $\forall_{S,O,M} \exists_{S2}$ sameStateExcept($S, S2, O$) $\wedge M =$ value($S2$, placement(O)).
(One can change the placement of object O in state S to M and construct a new state $S2$. $S2$ is not necessarily kinematically possible, but it is ontologically possible.)

DYN.2 $U \subset$objectsOf(H) $\Rightarrow$
$\exists_{H1}$ objectsOf($H1$) $= U \wedge$ sameHistoryOn($H1, H, U$).
(One can project a history H onto a subset U of its objects, getting a history $H1$.)

DYN.3 $\forall_{S,H}$ kinematic(S) $\wedge$ singleHist(H, S) $\Rightarrow$ dynamic(H).
(A history H consisting of a single kinematic state S is dynamic.)

DYN.4 $\forall_H$ dynamic(H) $\Rightarrow \exists_J$ unbounded(J) $\wedge$ dynamic(J) $\wedge$ historyPrefix(H, J).
(Any dynamic history H can be extended to an unbounded dynamic history J.)

DYN.5 dynamic(J) $\wedge$ historySlice($J1, J$) $\Rightarrow$ dynamic($J1$).
(Any temporal slice of a dynamic history is dynamic.)

DYN.6 dynamic($H1$) $\wedge$ dynamic($J2$) $\wedge$ hsplice($H1, J2, J$) $\Rightarrow$ dynamic(J).
(If $H1$ and $J2$ are dynamic and can be spliced together to form J, then J is dynamic. This excludes any kind of hysteresis.)

DYN.7 [dynamic(J) $\wedge$ throughout(J,isolated(UM, US)) $\wedge$
[$\forall_{O \in US}$ throughout($J, \neg^{\#}$freeMotion(O))] $\wedge$
objectsOf($J1$) $= UM \cup US \wedge$ sameHistoryOn($J, J1, UM \cup US$)
] $\Rightarrow$
dynamic($J1$).
(Discussed in text.)

DYN.8 [dynamic($J1$) $\wedge$ dynamic($J2$) $\wedge$
objectsOf(J) = objectsOf($J1$) $\cup$ objectsOf($J2$) $\wedge$
sameHistoryOn($J, J1$,objectsOf($J1$)) $\wedge$ sameHistoryOn($J, J2$,objectsOf($J2$)) $\wedge$
kinematic(J) $\wedge$ coherentGrasping($J1, J2$)] $\Rightarrow$
dynamic(J).
(Discussed in text.)

DYN.9 dynamic(J) $\Rightarrow$ kinematic(J).
(A dynamic history is kinematic.)

Table 17
Basic properties of dynamics: Axioms (continued).

DYN.10	$\text{history}(H) \wedge \text{dynamic}(H) \Rightarrow$ $\exists_{H1}\ \text{dynamic}(H1) \wedge \text{sameUntilEnd}(H, H1) \wedge \text{holds}(\text{end}(H1),\mathtt{freeGrasp})$.
DYN.11	$\text{dynamic}(H) \wedge \text{history}(H) \wedge \text{throughoutxSE}(H,\text{grasping}(O)) \Rightarrow$ $\exists_{H1}\ \text{dynamic}(H1) \wedge \text{sameUntilEnd}(H, H1) \wedge \text{holds}(\text{end}(H1),\text{grasping}(O))$.
DYN.12	$\text{holds}(S,\text{kinematicState}) \wedge \text{holds}(S,\mathtt{freeGrasp}) \Rightarrow$ $\exists_{H}\ \text{dynamic}(H) \wedge S = \text{start}(H) \wedge \text{startTime}(H) < \text{endTime}(H) \wedge$ $\text{throughout}(H,\mathtt{freeGrasp})$.
DYN.13	$\text{holds}(S,\text{kinematicState}) \wedge [\text{holds}(S,\text{grasping}(O)) \vee \text{holds}(S,\mathtt{freeGrasp})] \Rightarrow$ $\exists_{H}\ \text{dynamic}(H) \wedge S=\text{start}(H) \wedge \text{startTime}(H) < \text{endTime}(H) \wedge$ $\text{throughoutxSE}(H,\text{grasping}(O))$.
DYN.14	$[\text{kinematic}(HK) \wedge \text{holds}(\text{start}(HK),\text{grasping}(O)) \wedge$ $[\forall_{O1 \in \text{objectsOf}(H)}\ \text{parallelMotion}(O1, O, HK) \vee \text{motionless}(O1, HK)]] \Rightarrow$ $\exists_{H2}\ \text{startTime}(HK) = \text{startTime}(H2) < \text{endTime}(H2) \wedge \text{start}(H2) = \text{start}(HK) \wedge$ $\text{sameMotionOn}(H2, HK, \{O\}, 0) \wedge \text{throughoutxSE}(H2,\text{grasping}(O)) \wedge$ $\text{dynamic}(H2)$.
	(These are all discussed in the text.)

remain motionless. Clearly this is kinematically possible, given the condition COND.6. Axiom DYN.14 thus asserts that it is dynamically possible to carry the box along at least an initial segment of the specified trajectory, though the actual behavior of the objects may not be that of HK. The objects in the box may settle, or other objects may move, for reasons of their own.

4.6.3. Heaps and stability

We next address the issue of heaps and stable heaps (Table 18). Again, the discussion here is preliminary; it is adequate to the problem of carrying cargo in boxes, but a considerably richer and more powerful theory would be required to analyze the problem of carrying cargo piled in heaps on trays.

A *heap* is defined (definition HD.3) as in Section 3.6. As with dynamic histories, we do not give necessary and sufficient conditions for a collection of objects to be in a stable heaps; we just posit some of the axioms that are needed for the inference we are concerned with. A state is *stable* if every freely moving object is part of a stable heap supported by fixed and grasped objects.

Axiom H.1 asserts that a stable heap is a heap. Axiom H.2 asserts that if UH is a stable heap supported by US which remains motionless, and is isolated from all other objects then UH likewise remains motionless and remains a stable heap. Axiom H.3 asserts that if a set of objects is isolated and no object is moved by the agent, then eventually the set attains a stable state. Axiom H.4 asserts that if UH is a stable heap on US in state $S1$, and in $S2$ the objects in UH and US are in the same positions as in $S1$ and UH is isolated from everything except US in $S2$, then UH is a stable heap with supports US in $S2$. (That is, the positions of other objects that are not in contact with UH do not affect whether UH is stable.) Axiom HD.5, HD.6, and H.5 encode the default rule used to infer that the box does not fall over during loading (Section 3.3.6): If object OB is stably supported by motionless object OT, and the objects in set UC are always above the convex hull of the contact points of OB with OT, and $\{OB\} \cup UC$ are isolated from everything except OT, then by default OB remains motionless. Specifically, axiom HD.5 defines a history in which this default rule is violated as exhibiting anomaly 2. Axiom HD.6 defines a history H as satisfying the property “noAnomaly2” if no slice of H exhibits anomaly 2. Default rule H.5 states that dynamic histories by default satisfy the “noAnomaly2” property.

4.6.4. No upward motion

The final category of physical rules is the default rule that prevents the cargo from coming out of the top of the box. This follows our formulation in Section 3.7. Object O undergoes an *upward motion* with respect to object set US in H if, for every object $O1$ in US and for every coordinate system QC that can be “attached” to $O1$ at any time in H, the z-coordinate of the center of mass of O relative to QC is higher at the end of H than at the beginning of H. An upward motion of O in H relative to US is *anomalous* if O is part of a heap UH supported by US at the start of H and UH is isolated from everything except US during H. A history H has *no anomalous upward motions* if none of the objects in H have anomalous upward motions in any temporal slice of H. By default, a dynamic history has no anomalous upward motions.

4.7. Comprehension axioms for histories

We will need to reason that, if the agent is grasping an object, then he can move it along any “well-behaved” trajectory consistent with the laws of physics. The dynamic axioms DYN.1 through DYN.14, especially DYN.12, enumerate some conditions that are sufficient for a history to be physically possible. What we need additionally are axioms that assert that all these histories *exist;* specifically, that for any “well-behaved” (to be defined below) mathematical function from time to mappings, there exists a history in which object O follows that function. Such axioms are *comprehension* axioms for histories. There are two, not very different, ways to state these; either using first-order axiom schemas or using higher-order logic. There are subtle differences between the expressive power of these two, but nothing that affects our inferences here. In this paper, I will use first-order schemas to simplify the notational and sortal issues involved.

Table 18
Heaps and stability: Symbols and definitions.

Symbols:
connectedGroup(U: objectSet) $\rightarrow$ fluent[Bool].
allFree(U: objectSet) $\rightarrow$ fluent[Bool].
heap(UH, US: objectSet) $\rightarrow$ fluent[Bool].
stableHeap(UH, US: objectSet) $\rightarrow$ fluent[Bool].
stable(U: objectSet) $\rightarrow$ fluent[Bool].
anomaly2(H: history).
noAnomaly2(H: history).

Definitions:
HD.1 holds(S,connectedGroup(U)) $\Leftrightarrow$
$U \subset$objectsOf(S) $\wedge$
$\forall_{U1,U2}$ $U1 \neq \emptyset \wedge U2 \neq \emptyset \wedge U1 \cup U2 = U \Rightarrow$
$\exists_{O1 \in U1, O2 \in U2}$ holds(S,rccC$^{\#}$($\uparrow O1, \uparrow O2$)).
(A set of objects U is a connected group in state S if it cannot be divided into two spatially separated subsets $U1$ and $U2$).
HD.2 holds(S,allFree(U)) $\equiv \forall_{O \in U}$ holds(S,freeMotion(O)).
HD.3 holds(S,heap(UH, US)) $\Leftrightarrow$
holds(S,connectedGroup(UH) $\wedge^{\#}$ allFree(UH)) $\wedge$
$US \subset$objectsOf(S) $\wedge$ $US \cap UH = \emptyset$ $\wedge$
[$\forall_{OH \in UH, OS \in \text{objectsOf}(S)}$ holds(S, rccC$^{\#}$(OH, OS)) $\Rightarrow$ $OS \in UH \cup US$] $\wedge$
[$\forall_{OS \in US}\, \exists_{OH \in UH}$ holds(S,rccC$^{\#}$($\uparrow OS, \uparrow OH$))].
(Definition of a heap, as in Section 3.6.)
HD.4 holds(S,stable(U)) $\Leftrightarrow$
$\forall_{O \in U}$ holds(S,freeMotion(O)) $\Rightarrow$
$\exists_{UH,US \subset U}$ $O \in UH$ $\wedge$ holds(S,stableHeap(UH, US)) $\wedge$
$\forall_{OS \in US}$ $\neg$holds(S,freeMotion(OS)).
(A set of objects U is stable in state S if every mobile object in U is part of a stable heap supported by fixed or grasped objects.)
HD.5 anomaly2(H) $\equiv$
$\exists_{UC,OB,OT,S2}$
dynamic(H) $\wedge$ throughout(H,isolated($UC \cup \{OB\}$, OT) $\wedge^{\#}$ freeMotion(OB)) $\wedge$
sameStateOn(start(H),$S2$,$\{OB, OT\}$) $\wedge$ holds($S2$,stableHeap($\{OB\}$, $\{OT\}$)) $\wedge$
[$\forall_{O \in UC}$ throughout(H,above$^{\#}$($\uparrow O$, convexHull$^{\#}$($\uparrow OT \cap^{*} \uparrow OB$)))] $\wedge$
throughout(H,motionless(OT)) $\wedge$ $\neg$throughout(H,motionless(OB)).
HD.6 noAnomaly2(H) $\equiv \neg\exists_{H1}$ historySlice($H1, H$) $\wedge$ anomaly2($H1$).

Table 19
Heaps and stability: Axioms.

Axioms: (These are discussed in the text.)
H.1 holds(S,stableHeap(UH, US)) $\Rightarrow$ holds(S,heap(UH, US)) $\wedge$ $US \neq \emptyset$.
H.2 [dynamic(J) $\wedge$ holds(start(J),stableHeap(UH, US)) $\wedge$
throughout(J,isolated(UH, US)) $\wedge$ $\forall_{O \in US}$ motionless(J, O)] $\Rightarrow$
$\forall_{O \in UH}$ motionless(J, O) $\wedge$ throughout(J,stableHeap(UH, US)).
H.3 [dynamic(J) $\wedge$ unbounded(J) $\wedge$
throughout(J,isolated(UM, UF) $\wedge^{\#}$ allFree(UM)) $\wedge$
[$\forall_{O \in UF}$ motionless(J, O)]] $\Rightarrow$
$\exists_{J1}$ historySuffix($J1, J$) $\wedge$ throughout($J1$,stable($UF \cup UM$)).
H.4 holds($S1$,stableHeap(UH, UF)) $\wedge$ sameStateOn($S1, S2, UH \cup UF$) $\wedge$
holds($S2$,isolated(UH, UF)) $\Rightarrow$
holds($S2$,stableHeap(UH, UF)).

Reiterian Default Rule:
H.5 dynamic(J) : noAnomaly2(J) / noAnomaly2(J).

Table 20
Center of mass.

Symbol: centerOfMass(O) $\rightarrow$ pseudo.

Axioms:
CM.1 $\forall_{O:\text{object}}$ point(shape(centerOfMass(O)).
CM.2 $\forall_{O:\text{object}}$ shape(centerOfMass(O)) $\in$ convexHull(shape(O)).

The more serious question is what class of mathematical functions should be considered well-behaved. Note that this is, in general, a superset of the histories that are *physically* possible i.e. that satisfy dynamic(H). Rather, these are the histories that are, so to speak, conceptually possible. Therefore, the decision here is mostly a matter of the convenience of the theory developer.

Table 21
Default rule excluding anomalous upward motion.

Symbols:
upwardMotion(O object, US: objectSet, H: history).
anomalousUpwardMotion(O: object, H: history).
noAnomUpwardMotion(H: history).

Definitions:

UD.1 upwardMotion$(O, US, H) \equiv$
$\forall_{OS \in US, QC, SM}$ [source$(QC) = OS \wedge$ coordinateSystem$(QC) \wedge$ stateOf$(SM, H) \wedge$
value(SM,zAxis$^{\#}(\uparrow QC)$) = $\hat{z}$] $\Rightarrow$
value(end(H),zCoor$^{\#}$($\uparrow$centerOfMass(O),$\uparrow QC$)) $>$
value(start(H),zCoor$^{\#}$(centerOfMass($\uparrow O$),$\uparrow QC$)).

UD.2 anomalousUpwardMotion$(O, H) \equiv$
$\exists_{UH, US}$ $O \in UH \wedge$ holds(start(H),heap(UH, US)) $\wedge$
throughout(H,isolated$(UH, US) \wedge^{\#}$ allFree(UH)) $\wedge$ upwardMotion(O, US, H).

UD.3 noAnomUpwardMotion$(H) \equiv$
$\forall_{O \in \text{objects}(H), H1}$ historySlice$(H1, H) \Rightarrow \neg$anomalousUpwardMotion$(O, H1)$.

Reiterian Default Rule:
UP.1 dynamic(H) : noAnomUpwardMotion(H) / noAnomUpwardMotion(H).

There are two major constraints on the class of histories that we wish to enforce. The first, already discussed, is that we will require all histories to be continuous. The second, more subtle, can be stated in the following principle:

HCP.1 Let $h_1, h_2, h_3 \ldots$ be an infinite sequence of histories, such that h_i is a proper prefix of h_{i+1} for all i. Then there exists a uhistory j_∞ which is an extension of all the h_i.

The force of this principle is most clearly illustrated in terms of an example that violates it. Suppose that for each k, h_k is the history such that startTime$(h_k) = -1$, endTime$(h_k) = -1/k$, and for all T between startTime(h_k) and endTime(h_k), the placement of O at time T in h_k is a translation by distance $\sin(1/T)$ in the $\hat{x}$ direction. Then each h_k is a prefix of h_{k+1}, but there does not exist a j_∞ that subsumes them all, because the position of O does not converge to a limit at $T = 0$.

To avoid this, we require that histories satisfy a Lipschitz condition that, between times TA and TB, no point in any object moves a distance greater than `maxSpeed`$\cdot|TB - TA|$, where `maxSpeed` is a constant. It is easily shown that if $h_1, h_2 \ldots$ satisfy the Lipschitz condition and h_i is a prefix of h_{i+1}, then the limit history j_∞ exists and also satisfies the Lipschitz condition.

We chose to use this particular Lipschitz condition for reasons of simplicity. In the long run, imposing an attainable upper bound on speed could be problematic for a Newtonian theory; obviously it is not consistent with either Galilean or Einsteinian relativity or with the solution to some collision problems. There are many other possible condition that could be imposed instead for the same purpose. Keep in mind that there is no harm in imposing a very weak condition on histories, as long as it is sufficient to guarantee HCP.1.

The real reason that we need principle HCP.1 is to justify the conclusion (Lemma 1.5 of our formal proof) that it is always possible to *attempt* to carry out any given plan and to work on it until either it fails, it succeeds, or it cannot be continued. If we allow histories like those in our counter-example above, then the plan of moving an object along the path $f(t) = \sin(1/T)\hat{x}$ can be begun over the interval $[-1, -1/T]$ for every $T < 0$ but not over the interval $[-1, 0]$; it is difficult to define a semantics of planning in a way that accommodates this.

As stated above, HCP.1 involves quantifying over infinite sequences of histories. Rather than do that, we use axiom schema HC.3 below. Let $\Psi(H)$ be a property of histories H, and let $H0$ be a history satisfying Ψ. (Ψ may have associated some parameters X; in this case, each valuation on X determines a property of H.) Let $\Gamma(J1)$ be the property, "$J1$ is an extension of $H0$ and every proper prefix of $J1$ that extends $H0$ satisfies Φ". Then there exists a uhistory JM that is maximal with respect to Γ; that is, Γ holds on JM but not on any proper extension of JM.

Axiom schema HC.3 is approximately equivalent to principle HCP.1 in the following sense. On the one hand, HCP.1 plus the axiom of choice entails HC.3. Proof: Let us write $h_A < h_B$ if $H0$ is a prefix of h_A, h_A is a proper prefix of h_B, and $\Gamma(h_B)$; clearly this is a partial ordering. By Zorn's lemma, there exists a maximal set of histories that is linearly ordered under $<$. Let $\mathcal{H}$ be some such maximal totally ordered set. If the end times of the histories in $\mathcal{H}$ are bounded above, let TZ be the least upper bound of these end times; let h'_K be any history in $\mathcal{H}$ whose end time is greater than $TZ - 1/K$ and let h_K be the prefix of h'_K ending at time $TZ - 1/K$. If the end times of the histories in $\mathcal{H}$ are not bounded above, let h'_K be any history with end time greater than K, and let h_K be the prefix of h'_K with end time equal to K. In either case, it is immediate that $h_K < h_{K+1}$ so by HCP.1 there exists some j_∞ that extends all the h_K. If TZ is bounded, then the prefix of j_∞ that ends at TZ satisfies the conclusion of HC.3; if not, then j_∞ itself satisfies the conclusion of HC.3.

Conversely, HC.3 implies HCP.1, as long as h_K can be defined in terms of a first-order formula in H and K; the proof is immediate.

Axiom HC.2 asserts that any system of motions satisfying the Lipschitz condition constitutes a history. Here $\Psi(O, T, M)$ is a first-order formula that defines a mapping from object O and time T to rigid mapping M. Axiom HC.2 states that if, for

Table 22
Comprehension axioms on histories: Symbols and definitions.

Symbols:	
$\mathtt{maxSpeed} \rightarrow$ real.	
mapDist($M1, M2$: rigidMapping, O: object) $\rightarrow$ distance.	
Definitions:	
HCD.1	mapDist($M1, M2, O$) = $D \Leftrightarrow$ $[\exists_{P \in \text{shape}(O)}$ mappingDistance$(M1, M2, P) = D] \wedge$ $[\forall_{P \in \text{shape}(O)}$ mappingDistance$(M1, M2, P) \leqslant D]$.
HCD.2	Let $\Psi(O$: object, T: time, M: rigidMapping, $X)$ be a formula. Then Lipschitz$^{\Psi}(O, X, TS, TE)$ is defined to be the following formula: $\forall_{T1,T2:\text{time},M1,M2:\text{rigidMapping}}$ $TS \leqslant T1 \leqslant T2 \leqslant TE \wedge \Psi(O, T1, M1, X) \wedge \Psi(O, T2, M2, X) \Rightarrow$ mapDist$(M1, M2, O) \leqslant \mathtt{maxSpeed} \cdot (T2 - T1)$.
HCD.3	Let $\Phi(H$: history,$X)$ be a formula. AllPPs$^{\Phi}(H, X)$ is defined to be the formula: $\forall_{H1}$ historyProperPrefix$(H1, H) \Rightarrow \Phi(H1, X)$.

$T \geqslant TS$ and $O \in U$, Ψ defines a function from O and T to M that satisfies the Lipschitz condition, then there is a uhistory J that corresponds to Ψ; that is startTime$(J) = TS$, objectsOf$(J) = U$, and for every object $O \in U$ and time $T \geqslant TS$ the placement of O at time T in H is the value that satisfies $\Psi(O, T, H)$.

Axioms HC.1 and HC.2 give necessary and sufficient conditions for a system of object motions to constitute a history. For this domain, this is an almost complete characterization of the class of histories. A complete characterization would involve additionally:

- Necessary and sufficient conditions for the evolution of grasping relations over time.
- A uniqueness axiom stating that any two histories with the same time interval, the same objects, the same motions, and the same grasping relations are in fact the same history.
- A comprehension axiom on states, positing that any placement of a set of objects constitutes a state.

If we were to posit such a set of necessary and sufficient conditions, then quite a few of our axioms would in fact be consequences of these conditions, specifically T.1—T.5, K.4, K.5, G.1—G.2, DYN.1, DYN.2 and HC.3. We have not taken this approach for reasons of elaboration tolerance. Axioms like these are applicable to a wide range of temporal and physical theories; by contrast, the proposed necessary and sufficient conditions on histories apply only to the partial theory of rigid solid objects that we consider in this paper (Tables 20–22).

Finally, the comprehension axiom HC.4 is a variant of HC.3. HC.3 states that there is always a maximal history over a topologically closed history JM that extends H and satisfies the condition "Φ holds over all proper prefixes of JM". HC.4 states, in effect, that there is a maximal history either over a closed interval, or over an interval that is open on the right, that extends H and satisfies the condition "Φ holds over all closed prefixes (not necessarily proper) of JM". Since our ontology does not include histories over open time intervals, the statement of this is somewhat indirect. The point of this is to deal with time structures which branch at the end point of an interval. If the agent is attempting to keep a fluent like "grasping(O)" true, and, following an open history there is one end point that keeps grasping(O) true and one that makes it false, — i.e. the agent can choose to ungrasp(O) at the end of H — we need to deduce that the agent can choose the one that keeps it true, and can continue to keep it true through the continuum of such choices that must be made.

The formal statement of these axioms proceeds as follows. First, let define the distance between rigid mappings $M1$ and $M2$ relative to object O, mapDist$(M1, M2, O)$ as the maximum over all points P in shape(O) of the distance from $M1(P)$ to $M2(P)$ (definition HCD.1).

Let $\Psi(O$: object, T: time, M: rigidMapping, $X_1 \ldots X_k)$ be any open formula with free variables O, T, M, and optional additional free variables $X_1 \ldots X_k$ of any sort. For simplicity we will write the arguments $X_1 \ldots X_k$ as a single argument X.

Assume that for any given object O and parameter value X, Ψ implicitly defines a function $\Psi'_{O,X}(T)$ be from time T to a rigid mapping M. that is, $\Psi(O, T, M, X) \Leftrightarrow M = \Psi'_{O,X}(T)$. The formula Lipschitz$^{\Psi}(O, X, TS, TE)$ asserts that, for a given value of O and X, throughout the time interval TS, TE, the function $\Psi'_{O,X}(T)$ satisfies the Lipschitz condition that for any point P in shape(O), the distance from $[\Psi'_{O,X}(T1)](P)$ to $[\Psi'_{O,X}(T2)](P)$ is at most $\mathtt{maxSpeed} \cdot |T2 - T1|$.

The definition of the Lipschitz condition is given in definition HCD.2. Note that this includes the condition that Ψ defines a single-valued mapping (consider the case where $T2 = T1$). Axioms HC.1, HC.2, HC.3 can now be stated as in Table 23.

4.8. Executing plans

Since our objective is to validate `plan1`, we next need to define what it means for a plan to be a correct solution to a problem. That is, we need to define a semantics of plans. As our plans are partially specified and our model of time is continuous, this is not entirely an established theory.

Table 23
Comprehension axioms on histories.

Axioms:

HC.1 $\forall_{O,H,T1,T2,S1,S2}\ T1 \leqslant T2 \wedge \text{stateAt}(H,T1,S1) \wedge \text{stateAt}(H,T2,S2) \Rightarrow$
$\text{mapDist}(\text{value}(S1,\text{placement}(O)), \text{value}(S2,\text{placement}(O)), O) \leqslant$
$\texttt{maxSpeed} \cdot (T2 - T1)$.

Axiom HC.1 asserts that the Lipschitz condition holds on the formula,
"$\exists_S$ stateAt$(H,T,S) \wedge M = \text{value}(S, \text{placement}(O))$".

HC.2 Let Ψ and Lipschitz$^\Psi$ be as in HCD.2. Assume that the variables U, TS, and J do not appear free in Ψ. Then the following formula is an axiom:

$$\begin{aligned}&\forall_{U:\text{objectSet},TS:\text{time},X}\\ &\quad [\forall_{O\in U,TE:\text{time}}\ TS < TE \Rightarrow \text{Lipschitz}^\Psi(O,X,TS,TE)] \Rightarrow\\ &\quad \exists_{J:\text{uhistory}}\ \text{startTime}(J) = TS \wedge \text{unbounded}(J) \wedge \text{objectsOf}(J) = U \wedge\\ &\qquad \forall_{T,S,M}\ \text{stateAt}(J,T,S) \wedge \Psi(O,T,M,X) \Rightarrow \text{value}(S,\text{placement}(O)) = M.\end{aligned}$$

HC.3 Let $\Phi(H\text{: history},X)$ be an open formula with a free variable H of sort history and optionally other free variables X. Assume that variables JM and $H1$ are not free in Φ. Then the following is an axiom:

$$\begin{aligned}&\forall_{H,X}\ \text{AllPPs}^\Phi(H,X) \Rightarrow\\ &\quad \exists_{JM}\ \text{historyPrefix}(H,JM) \wedge \text{AllPPs}^\Phi(JM,X) \wedge\\ &\qquad \forall_{H1}\ \text{historyProperPrefix}(JM,H1) \Rightarrow \neg\text{AllPPs}^\Phi(H1,X).\end{aligned}$$

HC.4 Let $\Phi(H\text{: history},X)$ be as in HC.3. Then the following is an axiom:

$$\begin{aligned}&\forall_{H,X}\ \text{AllPPs}^\Phi(H,X) \Rightarrow\\ &\quad \exists_{JM}\ \text{historyPrefix}(H,JM) \wedge \text{AllPPs}^\Phi(JM,X) \wedge\\ &\qquad [\neg\exists_{J1}\ \text{sameUntilEnd}(J1,JM) \wedge \Phi(J1)] \vee\\ &\qquad [\Phi(JM) \wedge\\ &\qquad\ \forall_{H1}\ \text{historyProperPrefix}(JM,H1) \Rightarrow \neg\text{AllPPs}^\Phi(H1,X)\ ]].\end{aligned}$$

We will develop our theory top–down. A standard definition of the semantics of a partial plan is that the plan P *correctly achieves* a task T starting in state S if the following holds: for all H, if H is a history starting in S, and the agent attempts to execute P in H, then he succeeds in executing P in H and the execution of P constitutes a successful accomplishment of T. (This definition does not address the issues of knowledge preconditions or knowledge acquisition.)

What is meant by a *successful* execution of plan P in H is generally quite apparent from the form of P; representational systems for partial plans usually directly characterize a successful execution. What is meant by an *attempt* to execute P is often less obvious. For a TWEAK-style partial plan [4], for instance, we can define "attempts" as follows: a *beginning* of an execution of plan P is an execution of the first k steps of the plan under some ordering and variable binding consistent with the constraints. P is *attempted* in H if P is begun in H and cannot be continued, either because P is complete, or because there are no next steps whose preconditions are met.[16]

The first thought is to define an attempt to execute P as the initial segment of a complete execution of P; but under that definition of "attempt", the above definition of "correctness" become vacuous. Rather, we have to go in the opposite direction. We view the form of the plan as specifying what it means to "attempt" the plan, and as specifying when such an attempt constitutes a successful execution of the plan. Additionally, as mentioned above, the fact that we are dealing with real-valued time means that more care has to be taken than in similar theories over discrete time.

We proceed, then, as follows: The semantics of a plan P is specified in terms of four primitive predicates. The predicate "worksOn(P,H)" states that H constitute a partial or complete execution of P. The predicate "beginnable(P,S)" states that it is possible to initiate execution of P in S. The predicate "completion(P,H)" states that H constitutes a complete execution of P.

For example, consider the plan, `pHardBoil` = "Hard-boil egg `egg1`". The formula "beginnable(`pHardBoil`,$S1$)" is true if in $S1$ there is a pot $POT1$ with water on a burner $B1$ of the stove. The formula "completion(`pHardBoil`,H)" is true if the sequence "put `egg1` into $POT1$; turn the knob controlling $B1$; wait 12 minutes" is completely executed in H; and the formula "worksOn(`pHardBoil`,H)" is true if some initial segment of that sequence is executed in H.

The relations "attempts(P,J)" and "completes(P,H)" are defined in terms of the above three predicates. To simplify the process of plan definition, we require that our definitions of "attempts" and "completes" be coherent and well-behaved for *any* definition of "worksOn", "beginnable", and "completion", whether or not these satisfy any particular logical or temporal relations. For instance, intuitively one might think that if worksOn(P,H) holds, then worksOn$(P,H1)$ should hold for any

[16] This is actually weaker (more inclusive) than Chapman's semantics, because it requires only that at each there is some step in the plan whose predecessors have been executed and whose preconditions are met. By contrast, Chapman's semantics require that the preconditions are satisfied for every step whose predecessors have been executed. To achieve Chapman's semantics, it is necessary to define the beginning of the execution of a plan as the combination of the execution of some of the steps together with the choice of a next step to execute, among steps all of whose predecessors have been executed. Chapman's definition is computationally more tractable; it is verifiable in polynomial time, whereas verifying our definition is NP-hard.

initial segment of $H1$; or that if P is not beginnable in S then "worksOn(P, H)" should not hold over any history H that starts in S. However, we do not impose either of these conditions, or indeed any constraints whatever on these three base relations. Indeed, we could consistently specify a comprehension axiom, analogous to axiom P.2 of [8], which states that there exists a plan for any definition of these three predicates; we have not done that here because it is somewhat complex and we do not here need such an axiom. Achieving this in a theory of real-valued time requires a little care.

The predicate "attempts(P, J)" and some of the other predicates we will define, takes as argument a possibly unbounded uhistory J to deal with the case that the plan P goes into an infinite loop.

We posit a constant positive duration "reactionTime", which is the time required for the agent to "realize" that the completion criterion "completion(P, H)" has been met. Of course, this is often realistic, but that is not the actual reason that we are including this feature in our theory. (The small increase in realism would not be worth the complexity and inelegance.) Rather the point of including this is to deal coherently with cases where "completion(P, H)" holds over an interval of time that is open on the left (or more generally where it holds on a set of time points that does not contain its lower boundary). For instance, if the predicate "completion(P, H)" is defined as "the center of mass of O is more than 1 foot higher than the table", then there is no first time at which the predicate becomes true, so it would not be consistent to specify that the agent stops working on the plan as soon as the completion condition holds. Therefore, we define the predicate "reactComplete(P, H)" (the agent can achieve the completion criterion of P and react to it) as holding if H finishes at a time which is reactionTime greater than the greatest lower bound on the times at which "completion(P, H)" becomes true (PLD.1). Note it is possible that "completion(P, H)" will no longer be true by the time that "reactComplete(P, H)" becomes true, but that is not a problem; in such a case, "completes(P, H)" still holds (Tables 24, 25).

The imposition of a constant time delay on the completion of a plan has two further advantages in terms of achieving a coherent semantics. First, it eliminates the problem of defining "sequence($P1, P2$)" where $P1$ completes instantaneously; second, it eliminates the problems of loops that execute infinitely many iterations in finite time. The disadvantage of this is that we must now define "worksOn(P, H)" in such a way that the agent works on P for the duration reactionTime after P has met the completion criterion; if worksOn is not so defined, then "completes(P, H)" will not be achieved.

Our definitions now continue as follows: The predicate "baseExec(P, H)" (H is a strictly partial execution of P) holds if H is dynamic and worksOn(P, H) (PLD.2). The predicate "incompleteExec(P, H)" holds if baseExec(P, H) but not reactComplete(P, H) (PLD.3).

We need to define "attempt" in such a way that the agent continues to work on P as long as possible. There are two cases to be addressed here, corresponding to the two topologies of the right-hand side of a time interval. First, if the agent has been working on P over a closed time interval, and it is possible for him to continue working on P, then he does so. Second, if he has been working on P over an time interval that is open on the right, and it is possible for him to continue working on P over the closure of that interval, then he does so.

Thus, we define the following predicates. The predicate "beginsxE(P, H)" meaning "P begins over H except possibly at its end" holds if P is beginnable at the start of J and incompleteExec($P, H1$) holds over every proper prefix $H1$ of J (PLD.4). The predicate "begins(P, H)" holds if baseExec(P, H) holds as well (PLD.5).

A plan P is continuable at the end of H written "continuableEnd(P, H)" if beginsxE(P, H) and it would be possible to continue working on P at the last moment of H (PLD.6). A plan P is continuable after H, written continuable(P, H, Q) if it begins over some extension of H satisfying Q (PLD.7).

Plan P is *attempted* over H subject to isolation condition Q, (a) P is not beginnable at the start of H, and H consists of that single state; or (b) beginxE(P, H) but P cannot be continued at the end of H; or (c) P begins over H but is not continuable past H (PLD.8). P *completes* over H if P is attempted over H and reactComplete(P, H) (PLD.9).

Thus, if P completes over H, then P is beginnable at start(H); worksOn($P, H1$) holds over every initial segment $H1$ of H; and completion($P, H1$) holds over the initial segment $H1$ that ends a time reactionTime before the end of H.

A *problem* is a specification of a starting state, a success criterion, and an isolation condition. Plan P is a correct solution of problem R if the following holds: If H is a dynamic history starting in S satisfying the isolation condition, and P is attempted in H, then P is completed in H and H satisfies the success condition of R. For instance in our egg boiling example, we might posit that H satisfies the success conditions of the problem boilEgg if egg1 is hard-boiled at the end of H, and that it satisfies the isolation condition of H if no other object interferes with the burner knob, the pot, or the egg during H.

There is, however, a difficulty integrating our default rules with this definition of "correct solution". The default rules H.5 and UP.1 above support the following inference: if history h1 starts in state s1, plan1 is attempted in h1, and h1 satisfies the isolation conditions, then plan1 will be completed in h1 and h1 will satisfy the success condition of problem1. However, Reiterian default theory does not permit us to carry out universal abstraction and conclude that this condition holds for all histories H satisfying these conditions. Nor should it, at least in this instance. As we discussed in Section 3.3.5, it is not generally reasonable to assert that there is no history satisfying these conditions in which catapulting does not occur, since, with many sets of cargo objects, the agent can actually carry out the plan in such a way that catapulting occurs, if he so chooses. What we really want to say is that plan1 is a "generally correct" solution for problem1, meaning that *most* histories H that satisfy start(H) = start(problem1), isolationCondition(problem1,H) and attempts(plan1,H) also satisfy completes(plan1,H) and succeeds(problem1,H). This is an object-level relation between problems and plans (given some fixed probability distribution over histories) which is quite distinct from the default inference given above.

Table 24
Plan semantics: Sorts and symbols.

Sorts:
plan.
problem.

Symbols:
$\texttt{reactionTime} \rightarrow$ duration.
worksOn(P: plan, H: history).
beginnable(P: plan S: state).
completion(P: plan, H: history).
reactComplete(P: plan, H: history).
baseExec(P: plan, H: history).
beginsxE(P: plan, H: uhistory).
begins(P: plan, J: uhistory).
continuableEnd(P: plan, H:uhistory).
continuable(P: plan, J: uhistory).
attempts(P: plan, J: uhistory).
completes(P: plan, H: history).
startProblem(R: problem) $\rightarrow$ state.
isolationCondition(R: problem, J: uhistory).
succeeds(R: problem, H: history).

Table 25
Plan semantics.

Definitions:
PLD.1 reactComplete$(P, H) \equiv$
$\quad\forall_D\ D < \texttt{reactionTime} \Rightarrow$
$\quad\quad\exists_{TC,HC}\ TC <$endTime$(H) - D \wedge$ hSlice$(H,$startTime$(H),TC,HC) \wedge$
$\quad\quad\quad$completion(P, HC).
PLD.2 baseExec$(P, H) \equiv$
$\quad$dynamic$(H) \wedge$ beginnable$(P,$start$(H)) \wedge$ worksOn(P, H).
PLD.3 incompleteExec$(P, H) \equiv$ baseExec$(P, H) \wedge \neg$reactComplete(P, H).
PLD.4 beginsxE$(P, H) \equiv$
$\quad$beginnable$(P,$start$(H)) \wedge$ dynamic$(H) \wedge$
$\quad\forall_{H1}$ historyProperPrefix$(H1, H) \Rightarrow$ incompleteExec$(P, H1)$.
PLD.5 begins$(P, H) \equiv$ beginsxE$(P, H) \wedge$ baseExec(P, H).
PLD.6 continuableEnd$(P, H) \equiv$
$\quad\exists_{H1}$ sameUntilEnd$(H1, H) \wedge$ begins$(P, H1)$.
PLD.7 continuable$(P, H) \equiv$
$\quad\exists_{H1}$ historyProperPrefix$(H, H1) \wedge$ begins$(P, H1)$.
PLD.8 attempts$(P, J) \equiv$
$\quad[\neg$beginnable$(P,$start$(J)) \wedge$ singleHist$(J,$start$(J))] \vee$
$\quad[$beginsxE$(P, J) \wedge \neg$continuableEnd$(P, J)] \vee$
$\quad[$begins$(P, J) \wedge \neg$continuable$(P, J)]$.
PLD.9 completes$(P, H) \equiv$ attempts$(P, H) \wedge$ reactComplete(P, H).

Axiom:
PL.1 $0 < \texttt{reactionTime}$.

However, developing the requisite theory of measures of sets of histories is beyond the scope of this paper. We hope to return to this issue in future work, perhaps using a probabilistic logic along the lines of Bacchus [2].

It should be also noted that our treatment of isolation conditions is not actually quite what is wanted. The problem is that, in the above definition, a plan P vacuously satisfies the conditions for being a "correct" solution to a problem if P specifies that the agent should himself deliberately violate the isolation conditions. Or, even more cleverly, P could specify that if something goes wrong, then the agent should deliberately violate the isolation conditions. I have not found any adequate solution to this, and it may indeed be the case that, in the final analysis, boundary conditions for planning problems must be stated in terms of constraints on the starting state and not of constraints on the history. (Providing isolation from the actions of other agents would then entail calling on some richer theory of multi-agent interactions.) However, this concern does not affect the analysis in this paper. It certainly does not affect the validity of using isolation conditions in *physical* axioms such as H.2, H.4, and UD.2; it is only relevant to the question of the relation of plans to problems.

4.8.1. Control structures

In defining plan1, we use the standard programming language control structures "sequence" and "while". The semantics of these operators is specified in our theory by axioms which state how the predicates "beginnable", "worksOn", and "completion" are defined for a complex plan in terms of its components. The axioms are given in Table 26, and are self-explanatory.

Table 26
Semantics of plan control operators.

Symbols:
sequence($P1 \dots Pk$: plan) $\rightarrow$ plan.
if1(Q: fluent[Bool],P: plan) $\rightarrow$ plan. (Single-branch conditional).
while(Q: fluent[Bool],P: plan) $\rightarrow$ plan.

Axioms:

CTL.1	beginnable(sequence($P1$, $P2$),S) $\Leftrightarrow$ beginnable($P1$, S).
CTL.2	worksOn(sequence($P1$, $P2$),J) $\Leftrightarrow$ [worksOn($P1$, J) $\wedge \neg\exists_H$ historyProperPrefix(H, J) $\wedge$ completes($P1$, H)]$\vee$ $\exists_{H1,J2}$ hsplice($H1$, $J2$, H) $\wedge$ completes($P1$, $H1$) $\wedge$ worksOn($P2$, $J2$).
CTL.3	completion(sequence($P1$, $P2$),H) $\Leftrightarrow$ $\exists_{H1,H2}$ hsplice($H1$, $H2$, H) $\wedge$ completes($P1$, $H1$) $\wedge$ completion($P2$, $H2$).
CTL.4	beginnable(if1(Q, P),S) $\Leftrightarrow$ $\neg$holds(S, Q) $\vee$ beginnable(P, S).
CTL.5	worksOn(if1(Q, P),H) $\Leftrightarrow$ [holds(start(H),Q) $\wedge$ worksOn(P, H)] $\vee$ [$\neg$holds(start(H),Q) $\wedge$ throughout(H,`freeGrasp`)]. (Note: the condition throughout(H,`freeGrasp`) means simply that, if condition Q fails, the agent carries out a noop.)
CTL.6	completion(if1(Q, P),H) $\Leftrightarrow$ [holds(start(H),Q) $\wedge$ completion(P, H)] $\vee$ [$\neg$holds(start(H),Q) $\wedge$ endTime(H)=startTime(H)].
CTL.7	sequence($P_1, P_2 \dots P_k$) = sequence(P_1,sequence(P_2, ... sequence(P_{k-1}, P_k) ...)).
CTL.8	while(Q, P) = if1(Q,sequence(P,while(Q, P))).

Table 27
Axioms for primitive actions.

Symbols:
move(O: object,H: history) $\rightarrow$ plan.
waitUntil(Q: fluent[Bool]) $\rightarrow$ plan.

Axioms:

AC.1	beginnable(move(O, H),S) $\Leftrightarrow$ sameStateOn(S, start(H), $\{O\}$)
AC.2	worksOn(move(O, HT),H) $\Leftrightarrow$ $\exists_D$ D = startTime(H)-startTime(HT) $\wedge$ sameMotionOn(H, HT, $\{O\}$, D) $\wedge$ $\forall_{T,S}$ stateAt(H, $T+D$, S) $\Rightarrow$ [startTime(HT) $< T <$ endTime(HT) $\Rightarrow$ holds(S,grasping(O))] $\wedge$ [$T \geqslant$ endTime(HT) $\Rightarrow$ holds(S,`freeGrasp`)].
AC.3	completion(move(O, HT),H) $\Leftrightarrow$ $\exists_D$ D =startTime(H)$-$startTime(HT) $\wedge$ endTime(HT)+$D \leqslant$ endTime(H) $\wedge$ sameMotionOn(H, HT, $\{O\}$, D).
AC.4	$\forall_{Q,S}$ beginnable(waitUntil(Q),S).
AC.5	worksOn(waitUntil(Q),H) $\Leftrightarrow$ throughout(H,`freeGrasp`).
AC.6	$\forall_H$ completion(waitUntil(Q),H) $\Leftrightarrow$ $\exists_{T,S}$ stateAt(H, T, S) $\wedge$ holds(S, Q).

4.8.2. Primitive action: move and wait

There are two primitive actions in our theory:

- Move object O along the trajectory in history H. H is any history containing O; all that is significant about H is the trajectory that O follows in H.
- Wait until Boolean fluent Q becomes true.

Formally we consider these primitive actions to be of sort "plan". Their semantics is therefore given in terms of the predicates "beginnable", "worksOn", and "completion". The axioms are given in Table 27; they are mostly self-explanatory, but a few require some discussion.

Axiom AC.3 asserts that the completion condition for the action "Move O along trajectory HT" is met in history H if O executes the same motion in H as in HT subject to a time shift D and H continues at least as long as HT. However, our planning semantics requires that we define what it means to work on a plan for a time `reactionTime` *after* the completion condition is satisfied. (This is the disadvantage of positing a finite reaction time.) Axiom AC.2 reflects this. In history H, O is moved along trajectory HT if (a) the motion of O is H is the same as its motion in HT up until either the end time of H or the end time of HT, whichever comes first; (b) during the part of H up to the end time of HT, O is grasped; (c) if H continues past the end time of HT, then the agent is not doing anything in this final segment.

Table 28
Problem specification: Symbols.

Symbols:
`oTable1, oTable2` → object.
`oBox` → object.
`qInsideBox` → pseudoObject.
`qTopBox` → pseudoObject.
`rCuboid` → region. Empty cuboid inside the box.
`lCube,wCube,hCube` → distance.
`uCargo` → objectSet.
`u1` → objectSet. The set of movable objects (cargo and box).
`s1` → state. Initial state.
`s2` → state. Hypothetical state with the box on Table 2.
`maxCargoDiam` → distance. Maximum diameter of any cargo object.
`manipSpace1` → region. Free space above Table 1 for loading cargo into box.
`manipSpace2` → region.
    Free space from top of table 1 to top of Table 2 for carrying box.
`carryingPath` → history. Possible trajectory of box from Table 1 to Table 2.
loadingCount(D, L, W, H: distance) → integer.
    Conservative estimate of maximum number of cargo objects (see Section 3.3.2.)

4.9. Problem specification

We can now give the specification of our particular problem. There are, of course, many ways to formulate the problem. The choice of formulation involves a tradeoff between five desiderata: generality, that is, making the conditions as weak as possible; simplicity of the constraints; simplicity of the plan; standard form (e.g. describing regions in terms of specific geometric constraints rather than in terms of the existence of a path); and ease of constructing the proof of correctness. (Ease of proof is a desideratum because we are not really interested here in clever or deep object-level proofs, just in demonstrating the adequacy of the representation.) The specific choices we have made are largely arbitrary; we have tried to include generalizations that we felt were interesting and exclude those that are merely difficult.

The symbols (mostly constants) and axioms[17] here are given in Tables 28, 29 and 30. The symbols are explained in Table 28 where necessary. Axioms PR.1–PR.7 characterize the objects: The set `u1` includes the cargo and the box (PR.1), all of which are mobile (PR.2). PR.4 asserts that the box is box-shaped, and PR.3 defines the pseudo-objects associated with its inside and its opening. The two tables are fixed (PR.5 and PR.6). The distance `maxCargoDiam` is the maximum diameter of the cargo objects (PR.7).

Axioms PR.8 through PR.14 characterize the initial state `s1`. The state `s1` is kinematic (PR.8). The box opens vertically upward (PR.9). The inside of the box is empty (PR.10). Each cargo object constitutes a stable heap supported by `oTable1` (PR.11) and separated from every other cargo object (PR.12). No object in `u1` is overhung by any other object in `u1` or by `oTable1` (PR.13). The inside of the box is altogether above the convex hull of the contact points between the box and the table (PR.14); this is needed to make sure the box does not fall over (axiom H.4, Section 4.6.3).

Axioms PR.15 through PR.19 define `manipSpace1`, the region used to load objects into the box. We have constructed this definition so that each object can be lifted vertically, moved horizontally to a position over the inside of the box, and then lowered vertically into the box; in this part of the constraints, we have given priority to stating the constraints in simple geometric terms and keeping the proof of correctness easy. The region `manipSpace1` is defined as a region that is at least as high as the top of the box plus the height of any of the cargo objects (PR.15, PR.16). The table is nowhere higher than the top of the box (PR.17). For each cargo object O, `manipSpace1` includes the vertical prism whose x–y cross section is the convex hull of x–y projections of O and the box, and which extends vertically down from the top of `manipSpace1` a distance which at least the height of O (PR.18). `manipSpace1` also includes the region above every object O in `u1` up to the plane at the top of `manipSpace1` (PR.19).

Axioms PR.20 through PR.23 enforce the constraint needed to guarantee that the cargo objects will fit in the box and will not come out discussed in Sections 3.3.2 and 3.3.3. This puts an upper bound on the number of cargo objects as a function of the size of the cargo objects and the size of the inside of the box.

Axioms PR.24 through PR.27 characterize `s2`, a hypothetical state in which `oBox` sits on `oTable2`. (There is no need for `s2` to contain any objects other than `oBox` and `oTable2`.) State `s2` is kinematic (PR.24). `oTable2` is in the same position in `s2` as in s1 (PR.25). In `s2`, the box sits stably on `oTable2` (PR.26), the inside of the box is altogether above the table (PR.27), and, as in axiom PR.14, the inside of the box is altogether over the convex hull of the contact points between the box and the table (PR.28). PR.27 is required in order to achieve our goal (PR.35) that all the cargo objects are altogether above the table.

Axioms PR.29 through PR.31 characterize `carryingPath`, the trajectory of the box while being carried from `oTable1` to `oTable2`. (Here, by characterizing the space between these two boxes in terms of a trajectory rather than in absolute

[17] In the context of problem specification and plan specification, it is hard to distinguish between definitions and axioms, so we have lumped them all as axioms.

Table 29
Problem specification: Beginning.

PR.1	$\mathtt{u1} = \mathtt{uCargo} \cup \{\ \mathtt{oBox}\ \}$.
PR.2	$\forall_{O \in \mathtt{u1}}$ mobile(O).
PR.3	$\mathtt{oBox} =$ source$(\mathtt{qInsideBox})$ =source$(\mathtt{qTopBox})$.
PR.4	openBox$(\mathtt{oBox}, \mathtt{qInsideBox}, \mathtt{qTopBox})$.
PR.5	fixed$(\mathtt{oTable1})$.
PR.6	fixed$(\mathtt{oTable2})$.
PR.7	$\forall_{O \in \mathtt{uCargo}}$ diameter$(O) \leqslant \mathtt{maxCargoDiam}$.
PR.8	holds$(\mathtt{s1}, \mathtt{kinematicState})$.
PR.9	$\forall_P\ P \in$ value$(\mathtt{s1}, \mathtt{qTopBox}) \Rightarrow$ height$(P) =$ value$(\mathtt{s1}$,top$^\#(\uparrow\mathtt{qInsideBox}))$.
PR.10	holds$(\mathtt{s1}$,empty$(\mathtt{qInsideBox}))$.
PR.11	$\forall_{O \in \mathtt{u1}}$ holds$(\mathtt{s1}$,stableHeap$(\{O\}, \{\ \mathtt{oTable1}\}))$.
PR.12	$\forall_{O1,O2 \in \mathtt{u1}}$ holds$(\mathtt{s1}$,rccDC$^\#(\uparrow O1, \uparrow O2))$.
PR.13	$\forall_{O1 \in \mathtt{u1} \cup \{\mathtt{oTable1}\}, O2 \in \mathtt{u1}}\ \neg$holds$(\mathtt{s1}$,partlyAbove$^\#(\uparrow O1, \uparrow O2))$.
PR.14	holds$(\mathtt{s1}$,altogetherAbove$^\#(\uparrow\mathtt{qInsideBox}$, convexHull$^\#(\uparrow\mathtt{oBox} \cap^\# \uparrow\mathtt{oTable1})))$.
PR.15	$\forall_{O \in \mathtt{uCargo}}$ value$(\mathtt{s1}$,top$^\#(\uparrow O) -^\#$ bottom$^\#(\uparrow O)) \leqslant \mathtt{maxCargoHeight}$.
PR.16	$\forall_{O \in \mathtt{uCargo}}$ value$(\mathtt{s1}$,top$^\#(\uparrow\mathtt{oBox})) + \mathtt{maxCargoHeight} <$ top$(\mathtt{manipSpace1})$.
PR.17	value$(\mathtt{s1}$,top$^\#(\uparrow\mathtt{oTable1}) \leqslant$ value$(\mathtt{s1}$,top$^\#(\uparrow\mathtt{oBox}))$.
PR.18	$\forall_{O \in \mathtt{uCargo}, P}$ [xyProj$(P) \in$ value$(\mathtt{s1}$,convexHull$^\#$(xyProj$^\#(\uparrow O \cup \uparrow \mathtt{oBox}))) \wedge$ top$(\mathtt{manipSpace1}) - \mathtt{maxCargoHeight} \leqslant$ height$(P) \leqslant$ top$(\mathtt{manipSpace1})$] $\Rightarrow$ $P \in \mathtt{manipSpace1}$.
PR.19	$\forall_{O \in \mathtt{u1}, P, PO}$ holds$(\mathtt{s1}, PO \in^\# \uparrow O) \wedge$ pointAbove$(P, PO) \wedge$ height$(P) \leqslant$ top$(\mathtt{manipSpace1}) \Rightarrow$ $P \in \mathtt{manipSpace1}$.

Table 30
Problem specification: Conclusion.

PR.20	$\mathtt{rCuboid} \subset \mathtt{qInsideBox}$.
PR.21	cuboid$(\mathtt{rCuboid}, \mathtt{lCube}, \mathtt{wCube}, \mathtt{hCube})$.
PR.22	loadingCount$(D, L, W, H) = \lfloor L/2D \rfloor \cdot \lfloor W/2D \rfloor \cdot \lfloor H/2D \rfloor$.
PR.23	$\exists_N$ count$(\mathtt{uCargo}, N) \wedge$ $N \leqslant$ loadingCount$(\mathtt{maxCargoDiam}, \mathtt{lCube}, \mathtt{wCube}, \mathtt{hCube})$.
PR.24	holds$(\mathtt{s2}, \mathtt{kinematicState})$.
PR.25	sameStateOn$(\mathtt{s2}, \mathtt{s1}, \{\mathtt{oTable2}\})$.
PR.26	holds$(\mathtt{s2}$,stableHeap$(\{\mathtt{oBox}\},\{\mathtt{oTable2}\}))$.
PR.27	holds$(\mathtt{s2}$,altogetherAbove$^\#(\uparrow\mathtt{qInsideBox},\uparrow\mathtt{oTable2}))$.
PR.28	holds(s2,altogetherAbove$^\#(\uparrow\mathtt{qInsideBox}$, convexHull$^\#(\uparrow\mathtt{oBox} \cap^\# \uparrow\mathtt{oTable2})))$.
PR.29	sameStateOn(start$(\mathtt{carryingPath})$, $\mathtt{s1}$, $\{\mathtt{oBox}\}$).
PR.30	sameStateOn(end$(\mathtt{carryingPath})$,$\mathtt{s2}$,$\{\mathtt{oBox}\}$).
PR.31	$\forall_S$ stateOf$(S, \mathtt{carryingPath}) \Rightarrow$ verticalTilt(valueIn$(\mathtt{s1}$,placement$(\mathtt{oBox}))$, valueIn$(S$,placement$(\mathtt{oBox}))) = 0$.
PR.32	throughout$(\mathtt{carryingPath}, (\uparrow\mathtt{oBox} \cup^\# \uparrow\mathtt{qInsideBox}) \subset^\# \mathtt{manipSpace2})$.
PR.33	holds$(S$,isolFluent$(\mathtt{problem1})) \equiv$ $[\forall_{O:\text{object}}$ holds$(S$,rccC$^\#(\uparrow O, \mathtt{manipSpace1} \cup \mathtt{manipSpace2})) \Rightarrow$ $[O \in \mathtt{u1} \vee O = \mathtt{oTable1} \vee O = \mathtt{oTable2}]]$
PR.34	isolationConditions$(H, \mathtt{problem1}) \equiv$ throughout$(H$,isolFluent$(\mathtt{problem1}))$.
PR.35	succeeds$(\mathtt{problem1}, H) \equiv \forall_{O \in \mathtt{uCargo}}$ holds$(S$,altogetherAbove$(O, \mathtt{oTable2}))$.
PR.36	startProblem$(\mathtt{problem1}) = \mathtt{s1}$.

geometric terms, we have sacrificed stating constraints in normal form in favor of generality.) The trajectory carryingPath starts with oBox in its position in s1 (PR.29) and ends with the oBox in its position in s2 (PR.30), and oBox is held vertically upright throughout carryingPath (PR.31).

As we shall see in Section 4.10, the actual execution of plan1 need not follow carryingPath and need not end with the box at its position in s2. But the existence of this state and this trajectory guarantees the feasibility of the plan.

Axiom PR.32 characterizes the region manipSpace2 as including the swathe swept out by the box and its inside while moving it along carryingPath.

Finally, axioms PR.33 through PR.36 characterize the problem problem1. The isolation condition for problem1 is that no object other than cargo objects, oBox, and the two tables come into contact with either manipSpace1 or manipSpace2 during the execution of the plan. (PR.33, PR.34). PR.35 defines the goal of problem as achieving a state where all the objects in u1 are altogether above oTable2, and where they are in a stable state on oTable2. PR.36 defines the starting state of problem1 as s1.

4.10. Specification of plan1

Finally, we give the specifications of the plan. Our objective here is to define the plan as flexibly as possible. This flexibility has a number of advantages, depending on the application. For plan execution, it means that the fine details of

Table 31
Specification of `plan1`: Symbols.

Symbols:
`plan1` → plan.
`loadedCargo` → fluent[objectSet].
`unloadedCargo` → fluent[objectSet].
loadBox(U: fluent[objectSet], Q: pseudo, R: region) → plan.
`levelCount` → integer.
maxBottomHeight(N: integer) → distance.
loadBoxCondition(O: object, H: history, U: fluent[objectSet], Q: pseudo, R: region, S: state).
carryBox(OB: object, QI, QT: pseudo, UC: objectSet, OT: object, R: region) → plan.
boxLoadingPos(O: object, QI: pseudo) → fluent[Bool].
goodBoxTrajectory(H: history, OB: object, QIN, QT: pseudo, U: objectSet).
safeBoxTilt(S: state, QIN, QT: pseudo, O: object).
carryBoxConditions(H: history, OB: object, QIN, QT: pseudo, UC: objectSet, RM: region, OS: object, S: state).
moveTrajectory(H: history, O: object, U: objectSet, S: state, RM: region).
freeAbove(O: object, R: region) → fluent[Bool].
maxHeight(U) → fluent[distance].
bottom1(PS:pointSet, D:distance).

the plan (the order in which to load the objects, the placement of the objects in the box, the trajectories to follow) can be tailored to specific circumstances such as constraints on the way cargo objects should be packed, constraints on the way in which they should be moved, and limitations of the agent's manipulators. For plan recognition, the broader the definition of the plan, the more behaviors can be recognized as instances of the plan.

Axiom P1.1 gives the high-level description of `plan1`; the agent loads unloaded objects one by one into the box, waiting after each loading action until the objects have attained a stable state. The plan is composed of three basic actions:

- loadBox(`unloadedCargo,qInsideBox,manipSpace1`) is the action of loading some unloaded cargo object into the inside of the box, moving it through `manipSpace1`.
- carryBox(`oBox,qInsideBox,qTopBox,uCargo,oTable2,manipSpace2`) is the action of carrying box `oBox` with inside `qInsideBox` and opening `qTopBox` and with `uCargo` inside to its final position on `oTable2`, moving it through `manipSpace2`.
- waitUntil(stable(`u1` ∪ `oTable1`)) is the action of waiting until the cargo and the box have reattained a stable position on `oTable1`. The semantics of "waitUntil" are defined in Section 4.8.2.

Axioms P1.2 and P1.3 defines `loadedCargo` and `unloadedCargo` as the fluents whose value in any state is the set of cargo objects in the box/not in the box.

Axioms P1.4–P1.13 define the semantics of "loadBox" as a partial specification of a "move" action. Axioms P1.10 through P1.13 state that loadBox is executed (is beginnable/worked on/attains completion/attains failure) if in history H if an object is moved along a trajectory $H2$ that satisfies the "load box conditions". (There is no failure condition for "loadBox" for the same reason that there is no failure condition for "move"; a loadBox action fails if it is physically impossible to continue it.) Axiom P1.9 states that the trajectory $H2$ meets the load box conditions if the cargo object being loaded ends in an appropriate "box loading position" and it satisfies the predicate "moveTrajectory". Axioms P1.4–P1.7 defines "box loading position" as the constraint described in Section 3.3.2. (The wording of the condition in PL.6 is different, and in fact slightly less restrictive, than the condition in Section 3.3.2, just because the condition here is easier to state in our first-order theory.) Axiom P1.8 defines "moveTrajectory($H, O, UALSO, S, RM$)" as the constraints that history H starts in S; that in history H, O remains with within the manipulation space RM, and that H is kinematically consistent with all the objects outside $UALSO$ remaining motionless. The set $UALSO$ is the set of objects that should be "brought along" by the move; specifically, the cargo objects while the box is being moved.

Note that, although we specified the situation `s1` so that there was room to load each cargo object by moving it up vertically, then horizontally, then down vertically, the plan does not require that the loading actually be carried out that way.

Similarly, axioms P1.14–P1.22 characterize "carryBox" as a partial specification of a move. Axioms P1.19–P1.22 state that an execution of carryBox is an execution of a move that satisfies the "carry box conditions". Axiom P1.18 defines "carryBox-Conditions" as the constraints that were placed on the final situation s2 in axioms PR.25–PR.29 plus the "moveBoxTrajectory" condition discussed above plus the condition from Section 3.3.3 that the box is never tilted too far from the vertical. This last condition on tilting is defined in axioms P1.16 and P1.17. Axioms P1.14 and P1.15 define the maximum and minimum height of a set of objects (Tables 31–33).

Table 32
Specification of `plan1`: Definition of loadBox.

Axioms:	
P1.1	$\texttt{plan1} =$ $\text{sequence}(\text{while}(\texttt{unloadedCargo} \neq^{\#} \emptyset,$ $\text{sequence}(\text{loadBox}(\texttt{unloadedCargo},\texttt{qInsideBox},\texttt{manipSpace1}),$ $\text{waitUntil}(\text{stable}(\texttt{u1} \cup \{\texttt{oTable1}\}))))$ $\text{carryBox}(\texttt{oBox},\texttt{qInsideBox},\texttt{qTopBox},\texttt{uCargo},\texttt{oTable2},\texttt{manipSpace2})).$
P1.2	$O \in \text{value}(S,\texttt{loadedCargo}) \Leftrightarrow O \in \texttt{uCargo} \wedge \text{holds}(S,\uparrow O \subset^{\#} \uparrow \texttt{qInsideBox}).$
P1.3	$O \in \text{value}(S,\texttt{unloadedCargo}) \Leftrightarrow O \in \texttt{uCargo} \wedge \neg \text{holds}(S,\uparrow O \subset^{\#} \uparrow \texttt{qInsideBox}).$
P1.4	$\text{holds}(S,\text{freeAbove}(O,R)) \Leftrightarrow$ $\neg \exists_{O1 \in \text{objectsOf}(S)}\ O1 \neq O \wedge \text{holds}(S,\text{partlyAbove}^{\#}(\uparrow O1, \uparrow O)) \wedge$ $\text{holds}(S,\text{rccO}^{\#}(\uparrow O1, R)).$
P1.5	$\texttt{levelCount} = \lfloor \texttt{lCube}/\ 2^{*}\ \texttt{maxCargoDiam} \rfloor * \lfloor \texttt{wCube}/\ 2^{*}\ \texttt{maxCargoDiam} \rfloor.$
P1.6	$\text{maxBottomHeight}(N) = 2 \cdot \texttt{maxCargoDiam} * \lfloor N/\texttt{levelCount} \rfloor.$
P1.7	$\text{holds}(S,\text{boxLoadingPos}(O,QI)) \Leftrightarrow$ $\text{holds}(S,\texttt{kinematic}) \wedge \text{holds}(S,\text{freeAbove}(O,\texttt{manipSpace1})) \wedge \text{holds}(S,\uparrow O \subset^{\#} \uparrow QI) \wedge$ $[\exists_{O1 \in \texttt{u1}}\ \text{holds}(S, \text{rccEC}^{\#}(\uparrow O1, \uparrow O))] \wedge$ $\forall_{N}\ \text{count}(\text{value}(S,\texttt{loadedCargo}),N) \Rightarrow$ $\text{holds}(S,\text{height}^{\#}(\uparrow \text{centerOfMass}(O)) \leqslant^{\#}$ $\text{bottom}^{\#}(\texttt{rCuboid}) +^{\#} \text{maxBottomHeight}^{\#}(N) +^{\#} \texttt{maxCargoDiam}).$
P1.8	$\text{moveTrajectory}(H,O,UALSO,S,RM) \Leftrightarrow$ $S = \text{start}(H) \wedge \text{throughout}(H,\uparrow O \subset^{\#} RM) \wedge$ $\forall_{O1 \in \text{objectsOf}(H)}\ O1 \notin \{O\} \cup UALSO \Rightarrow$ $\text{motionless}(H,O1) \wedge \text{throughoutxSE}(H, \text{rccDC}^{\#}(O,O1)).$
P1.9	$\text{loadBoxConditions}(O,H,U,QI,RM,S) \Leftrightarrow$ $O \in \text{value}(S,U) \wedge \text{holds}(\text{end}(H),\text{boxLoadingPos}(O,QI)) \wedge$ $\text{moveTrajectory}(H,O,\emptyset,S,RM).$
P1.10	$\text{beginnable}(\text{loadBox}(U,QI,RM),S) \Leftrightarrow$ $\exists_{O,H}\ \text{loadBoxConditions}(O,H,U,QI,RM,S).$
P1.11	$\text{worksOn}(\text{loadBox}(U,QI,RM),H) \Leftrightarrow$ $\exists_{O,H2}\ \text{loadBoxConditions}(O,H2,U,QI,RM,\text{start}(H)) \wedge$ $\text{worksOn}(\text{move}(O,H2),H).$
P1.12	$\text{completion}(\text{loadBox}(U,QI,RM),H) \Leftrightarrow$ $\exists_{O,H2}\ \text{loadBoxConditions}(O,H2,U,QI,RM,\text{start}(H)) \wedge$ $\text{completion}(\text{move}(O,H2),H).$
P1.13	$\neg \text{failure}(\text{loadBox}(U,QI,RM),H).$

Table 33
Specification of `plan1`: Definition of carryBox.

P1.14	$\forall_{U:\text{objectSet},\,S:\text{state}}$ $[\forall_{O \in U,P}\ \text{holds}(S,P \in^{\#} \uparrow O) \Rightarrow \text{height}(P) \leqslant \text{value}(S,\text{maxHeight}(U))] \wedge$ $[\exists_{O \in U,P}\ \text{holds}(S,P \in \uparrow O) \wedge \text{height}(P) = \text{value}(S,\text{maxHeight}(U))].$
P1.15	$\text{bottom1}(R,D) \equiv$ $[\forall_{P \in R}\ \text{height}(P) \geqslant D] \wedge [\exists_{P \in R}\ \text{height}(P) = D].$
P1.16	$\text{goodBoxTrajectory}(H,OB,QIN,QTOP,UCARGO) \equiv$ $\forall_{O \in UCARGO,S,PHI}\ \text{stateOf}(S,H) \wedge$ $PHI = \text{verticalTilt}(\text{value}(\text{start}(H),\text{placement}(OB)),\ \text{value}(S,\text{placement}(OB))) \Rightarrow$ $\text{safeBoxTilt}(PHI,\text{start}(H),QIN,QTOP,O).$
P1.17	$\forall_{D1}\ \text{holds}(S,\text{bottom1}^{\#}(\uparrow QTOP,D1)) \Rightarrow$ $[\text{safeBoxTilt}(PHI,S1,QIN,QTOP,O) \Leftrightarrow$ $0 \leqslant PHI < \pi/2 \wedge$ $(D1\text{-value}(S,\text{height}^{\#}(\uparrow \text{centerOfMass}(O))) \cdot \cos(PHI) >$ $\text{diameter}(O) + \text{diameter}(\text{value}(S,\text{xyProj}^{\#}(\uparrow QTOP \cup \uparrow QIN))) \cdot \sin(PHI)].$
P1.18	$\text{carryBoxConditions}(H,OB,QIN,QTOP,UCARGO,RM,OS2,S) \Leftrightarrow$ $\text{holds}(S,\text{freeSpace}(\text{value}(\text{end}(H),\uparrow OB))) \wedge$ $\text{holds}(\text{end}(H),\text{altogetherAbove}(\uparrow QIN,\uparrow OS2)) \wedge$ $\text{holds}(\text{end}(H),\text{altogetherAbove}^{\#}(\uparrow QIN,\text{convexHull}^{\#}(\uparrow OB \cap^{\#} \uparrow OS2))) \wedge$ $\text{moveTrajectory}(H,O,UCARGO,S,RM) \wedge$ $\text{goodBoxTrajectory}(H,OB,QIN,QTOP,UCARGO).$
P1.19	$\text{beginnable}(\text{carryBox}(OB,QIN,QTOP,UCARGO,OS2,RM),S) \Leftrightarrow$ $\exists_{H}\ \text{carryBoxConditions}(H,OB,QIN,QTOP,UCARGO,RM,OS2,S).$
P1.20	$\text{worksOn}(\text{carryBox}(OB,QIN,QTOP,UCARGO,OS2,RM),H) \Leftrightarrow$ $\exists_{H2}\ \text{carryBoxConditions}(H2,OB,QIN,QTOP,UCARGO,OS2,RM,\text{start}(H))$ $\wedge\ \text{worksOn}(\text{move}(OB,H2),H).$
P1.21	$\text{completion}(\text{carryBox}(OB,QIN,QTOP,UCARGO,OS2,RM),H) \Leftrightarrow$ $\exists_{H2}\ \text{carryBoxConditions}(H2,OB,QIN,QTOP,UCARGO,OS2,RM,\text{start}(H))$ $\wedge\ \text{completion}(\text{move}(OB,H2),H).$
P1.22	$\neg \text{failure}(\text{carryBox}(OB,QIN,QTOP,UCARGO,OS2,RM),H).$

5. Sketch of proof

We now can prove the following result: Given all the above, and given that for uhistory j1,

J1.1 start(j1) = startProblem(problem1) = s1.
J1.2 isolationCondition(problem1, j1)
J1.3 attempts(plan1, j1)

one can infer by default that completes(plan1, j1) and succeeds(problem1, j1).

A fully detailed account of the proof is given in the online appendix http://cs.nyu.edu/faculty/davise/box-proof.pdf. The overall structure of the plan is a straightforward projection, though the details involve a lot of definition hunting, and a fair amount of the kind of fiddly argumentation characteristic of the analysis of continuous functions over real-valued time.

To analyze the loading loop, we use the following loop invariant: The cargo objects outside the box and the box itself are in the same position as in s1. The cargo objects inside the box are in a stable position, and satisfy the following packing constraint: Let K be the number of cargo objects in the box, let H be the greatest height of the center of mass of any cargo object in the box, and let maxCargoDiam, lCube and wCube be as defined in axioms PR.7 and PR.21. Then

$$K \geqslant \left\lfloor \frac{\texttt{lCube}}{2\cdot\texttt{maxCargoDiam}} \right\rfloor \cdot \left\lfloor \frac{\texttt{wCube}}{2\cdot\texttt{maxCargoDiam}} \right\rfloor \cdot \left\lfloor \frac{H}{2\cdot\texttt{maxCargoDiam}} \right\rfloor.$$

The main steps of the proof are as follows:

- The loop invariant holds in s1.
- Given the above loop invariant, at the beginning of each iteration, there will exist a loading action that satisfies the box loading constraint.
- Any move that satisfies the box loading constraint will execute successfully, and that when it is finished the new object will be inside the box and the above packing condition will hold.
- After a loading action is complete and the object being loaded is released, the cargo will settle into a new stable position with the box remaining fixed, and the cargo objects all remaining inside the box.
- Once the cargo has settled into a stable condition, the loop invariant is satisfied.
- The loading loop terminates because the number of unloaded cargo objects decreases on each iteration.
- At the completion of the loading loop, there exists a move that satisfies the box carrying constraint.
- If a move satisfies the box carrying constraint, then it can be executed successfully, and will result in a state that satisfies the goals of the problem. In particular, the cargo objects do not come out of the box while it is being carried.

It may be noted that we have been careful to avoid the use of default rules until the very last step of the proof. The last lemma preceding the final theorem

Lemma:
start(J) = s1 $\wedge$ attempts(plan1, J)$\wedge$
throughout(J, isolFluent) $\wedge$ noAnomaly2(J) $\wedge$ noAnomUpwardMotion(J) $\Rightarrow$
completes(plan1, J) $\wedge$ succeeds(problem1, J)

has been proved using purely first-order logic. Therefore, this lemma has been validly proved even if some problem arises with the default rules.

6. Conclusion

Among all these trees, it is easy to lose sight of the forest. What we have accomplished is this: We have developed a theory that is capable of justifying a commonsensically obvious inference about using boxes to carry cargo. The inference requires only qualitative constraints about the shapes and physical characteristics of the objects involved. The theory is designed to be elaboration tolerant and consistent with Newtonian physics; it contains no features that get in the way of extending it to cover both other commonsense inferences in the domain and precise calculations based on Newtonian mechanics. As discussed in the introduction, our formulation of the physical laws used in this inference entirely avoids the analysis of forces, and almost entirely avoids the use of axioms that use differential time (the only exceptions are axioms K.5 and DYN.12).

The theory, the boundary conditions that define the problem, and the representation of the plan, are certainly more complicated than one would at first have supposed necessary for such an obvious inference in such a simple domain. But on careful consideration, it seems clear that a commonsense understanding of the domain and of this inference involves all, or nearly all, of the sorts that we have defined, and is aware of all, or nearly all, of the potential "bugs" that we have enumerated in Section 3.3. Therefore it seems reasonable to say that the complexity of our theory is mostly a reflection of the complexity of the domain and the sophistication of a commonsense understanding, and only in small part an artifact of the awkwardness of fitting this kind of commonsense reasoning to the limitations of deductive inference in first-order logic.

Certainly, the theory here is much more numerically precise than an actual commonsense understanding; no one would claim that any actual commonsense reasoner thinks about conditions PR.23 and PR.24 or knows that these conditions are sufficient to ensure that the box can be loaded without fear of overflowing. But a commonsense understanding *is* aware of something quite similar: namely, that if care is taken to load the box from bottom up, the space will be used reasonably efficiently; and that, if objects are small and not too numerous, a reasonably efficient packing of the box will succeed in fitting them all inside the box. Moreover, the commonsense understander has a "feel" for how small the objects should be, how few they should be, and how much care needs to be taken in packing them. The precise numerical constraints PR.23, PR.24 here are the closest we have been able to come to representing the knowledge that constitutes this "feel". The numerical constraints are not, I would argue, as far from the "feel" as it might seem at first; and they are certainly no further than what can be expressed in any other notation that I know of. Despite the scorn that is often heaped on the very idea that symbolic representations could be acceptable cognitive models for spatial knowledge,[18] no other representational system, especially "diagrammatic" representations, comes anything like as close to capturing the critical cognitive ability to represent and reason about qualitative spatial and physical information.

Establishing the consistency of this large and complex theory, even aside from the default rules, is certainly a concern. The major crux is likely to be finding a class of dynamic histories that satisfies both the existence and closure axioms DYN.2–DYN.14 and also the rules for heaps H.2 to H.4. Another problem is that there is an inherent tension between DYN.6, which excludes any kind of hysteresis, and default rule UP.1, which has hysteresis built in (the motions possible to objects in the heap at one time depend on their positions relative to the support at a different time.) I don't think that these are actually inconsistent, but it is certainly possible that they suffice to rule out important forms of "settling" if the box is tilted while being carried.

The work in this paper is only a first step in the analysis of commonsense knowledge about solid objects. The most important open problems in this analysis, which we hope to address in future work, are:

- Incorporating a probabilistic theory or some other theory of relative likelihoods.
- Analyzing the unloading of the box.
- Merging this theory with the Newtonian theory of forces.
- Developing a more realistic model of manipulation.
- Extending the theory of the stability of heaps under perturbations (motions of the supports, contacts with external objects, and impacts of external objects).

References

[1] E.W. Adams, The foundations of rigid body mechanics and the derivation of its laws from those of particle mechanics, in: L. Henkin, P. Suppes, A. Tarski (Eds.), The Axiomatic Method — With Special Reference to Geometry and Physics, North-Holland, Amsterdam, 1959, pp. 250–265.
[2] F. Bacchus, Representing and Reasoning with Probabilistic Knowledge: A Logical Approach to Probabilities, MIT Press, 1990.
[3] B. Bennett, A.G. Cohn, P. Torrini, S.M. Hazarika, Describing rigid body motions in a qualitative theory of spatial regions, in: AAAI-00, pp. 503–509.
[4] D. Chapman, Planning for conjunctive goals, Artificial Intelligence 32 (1987) 333–378.
[5] E. Charniak, Statistical Language Learning, MIT Press, 1993.
[6] E. Davis, A logical framework for commonsense predictions of solid object behavior, Int. Journal of AI in Engineering 3 (3) (1988) 125–140.
[7] E. Davis, The naive physics perplex, AI Magazine 19 (4) (Winter 1998) 51–79.
[8] E. Davis, L. Morgenstern, A first-order theory of communication and multi-agent plans, Journal of Logic and Computation 15 (5) (2005) 701–749.
[9] E. Davis, The expressivity of quantifying over regions, Journal of Logic and Computation 16 (2006) 891–916.
[10] E. Davis, Physical reasoning, in: Frank van Harmelen, Vladimir Lifschitz, Bruce Porter (Eds.), The Handbook of Knowledge Representation, Elsevier, Oxford, 2007, pp. 597–620.
[11] J. de Kleer, Multiple representations of knowledge in a mechanics problem solver, in: IJCAI-77, pp. 299–304.
[12] B. Faltings, Qualitative kinematics in mechanisms, in: Proc. IJCAI-87, pp. 436–443.
[13] K. Forbus, Spatial and qualitative aspects of reasoning about motion, in: Proc. AAAI-80, pp. 170–173.
[14] K. Forbus, P. Nielsen, B. Faltings, Qualitative spatial reasoning: The CLOCK project, Artificial Intelligence 51 (1991) 417–471.
[15] A. Gelsey, Automated reasoning about machines, Artificial Intelligence 74 (1995) 1–53.
[16] S. Hanks, D. McDermott, Nonmonotonic logic and temporal projection, Artificial Intelligence 33 (1987) 379–412.
[17] P. Hayes, The naive physics manifesto, in: D. Michie (Ed.), Expert Systems in the Microelectronic Age, Edinburgh University Press, Edinburgh, 1979.
[18] S. LaValle, Automated Planning, Cambridge University Press, 2006.
[19] D. Lenat, The voice of the turtle: Whatever happened to AI?, AI Magazine 29 (2) (2008) 11–22.
[20] V. Lifschitz, Formal theories of action, in: F. Brown (Ed.), The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop, Morgan Kaufmann, 1987.
[21] D. Macaulay, The Way Things Work, Houghton Mifflin, 1988.
[22] J. McCarthy, Programs with common sense, in: Proc. Symposium on Mechanization of Thought, vol. 1, London, 1959.
[23] J. McCarthy, Programs with common sense, in: M. Minsky (Ed.), Semantic Information Processing, MIT Press, Cambridge, MA, 1968, pp. 403–418.
[24] J. McCarthy, Circumscription — A form of nonmonotonic logic, Artificial Intelligence 13 (1980) 27–39.
[25] J. McCarthy, Elaboration tolerance, in: 4th International Symposium on Logical Formalizations of Commonsense Reasoning, 1998.
[26] D. McDermott, Tarskian semantics, or no notation without denotation!, Cognitive Science 2 (1978) 277–282.
[27] D. McDermott, A temporal logic for reasoning about processes and plans, Cognitive Science 6 (1982) 101–155.
[28] D. McDermott, The 1998 AI planning systems competition, AI Magazine 21 (2000).

[18] For instance, Waltz [37], p. 398 writes that "It was widely believed that logic could successfully model images and scenes, even though the baroque improbability of that effort should have long been clear to everyone who read Pat Hayes' Naive Physics Manifesto".

[29] P. Nielsen, A qualitative approach to mechanical constraint, in: Proc. AAAI-88, pp. 270–274.
[30] D.A. Randell, Z. Cui, A.G. Cohn, A spatial logic based on regions and connection, in: Third International Conference on Principles of Knowledge Representation and Reasoning, 1992, pp. 165–176.
[31] R. Reiter, A logic for default reasoning, Artificial Intelligence 13 (1980) 81–132.
[32] R. Reiter, Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems, MIT Press, Cambridge, MA, 2001.
[33] M. Shanahan, Solving the Frame Problem, MIT Press, 1997.
[34] T. Stahovich, R. Davis, H. Shrobe, Qualitative rigid-body mechanics, Artificial Intelligence 119 (2000) 19–60.
[35] D. Stewart, Existence of solutions to rigid body dynamics and the paradoxes of Painlevé, Comptes Rendus de l'Academie des Sciences, Ser. I 325 (1997) 689–693.
[36] D. Stewart, Rigid-body dynamics with friction and impact, SIAM Review 41 (1) (2000) 3–39.
[37] D. Waltz, Cognitive and computation models: Section introduction, in: J. Glasgow, H. Narayanan, B. Chandrasekaran (Eds.), Diagrammatic Reasoning: Cognitive and Computational Perspectives, AAAI Press, 1995, pp. 397–401.