Proceedings, Eleventh International Conference on Principles of Knowledge Representation and Reasoning (2008)

Cartesian Situations and Knowledge Decomposition in the Situation Calculus

Ronald P. A. Petrick
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB, Scotland, UK
rpetrick@inf.ed.ac.uk

Abstract

We formalize the notion of a Cartesian situation in the situation calculus, a property that imposes strong structural conditions on the configuration of a set of possible worlds. Focusing on action theories that use the Scherl and Levesque account of knowledge and action, we show how Cartesian situations give rise to a set of decomposition properties for simplifying epistemic formulae (in particular, certain disjunctive and existentially quantified formulae) into equivalent components that only mention fluent literals. Moreover, we describe certain expressive classes of action theories that preserve the Cartesian property through action. This work also offers the possibility of identifying action theories that can be compiled into alternative accounts of knowledge that have similar representational restrictions, but do not use possible worlds.

Introduction and Motivation

Agents that act in dynamic worlds often do so with incomplete information about their environments. Agents that also sense the world also have the ability to gather new information and extend their knowledge about the world. As a result, logical accounts that model the behaviour of agents in such settings, for instance for the purpose of planning or high-level control, therefore require the ability to reason effectively about knowledge and action.

The problem of reasoning about knowledge and action has been widely studied and is relatively well understood. In the situation calculus (McCarthy and Hayes 1969), the standard approach to this problem due to Moore (1985) views knowledge in terms of an accessibility relation over a set of possible worlds (Hintikka 1962). Scherl and Levesque (1994; 2003) adapt this approach in Reiter's version of the situation calculus (Reiter 2001) and provide a solution to the frame problem (McCarthy and Hayes 1969) for knowledge change.

While the Scherl and Levesque approach is representationally rich, it is computationally problematic: reasoning about whether n atomic formulae are known potentially involves reasoning about 2^n possible worlds. As a result, more recent approaches have explored alternate representations of knowledge and action that do not require an accessibility relation and possible worlds (e.g., (Funge 1998), (Demolombe and Pozos Parra 2000), (Soutchanski 2001), (Son and Baral 2001), (Petrick and Levesque 2002), (Liu and Levesque 2005), (Vassos and Levesque 2007)). To achieve more tractable reasoning, many of these accounts sacrifice expressiveness by restricting the types of knowledge that can be represented. A common approach is to limit disjunctive knowledge and model an agent's knowledge state by sets of fluents that are known to be true or known to be false.

How do the limitations of these alternate representations relate to the standard Scherl and Levesque account of knowledge? Can we restrict Scherl and Levesque action theories to bring about similar representations? What must we give up to do so? McCarthy (1979a; 1979b) points out that a Cartesian product structure on a space of possible worlds introduces an independence property that removes dependencies between individual domain fluents. In this paper we formalize the notion of a *Cartesian situation*, a property inspired by the ordinary Cartesian product over sets, that imposes strong structural conditions on the configuration of a set of possible worlds. We show that Cartesian situations give rise to a collection of "decomposition" properties that let us simplify epistemic formulae—in particular, certain disjunctive and existentially quantified formulae—into equivalent components that only mention fluent literals. We also describe certain expressive classes of action theories that preserve the Cartesian property through action.

Treating knowledge in terms of fluent literals closely corresponds to the restrictions made by some of the alternative accounts of knowledge. While the formal properties we establish will help us better understand the nature of Cartesian situations, this work also offers the possibility of identifying standard action theories that can be compiled into alternative representations supporting more tractable reasoning.

This paper is organized as follows. We begin by describing the situation calculus and the Scherl and Levesque model of knowledge. We then define Cartesian situations and establish a set of decomposition properties for particular types of knowledge formulae. We identify certain classes of action theories that preserve the Cartesian property through action, and give examples of such theories. We close with a discussion of this work and its connection to related approaches.

Situation Calculus

The situation calculus (as described in (Reiter 2001)) is a first-order, many-sorted language (with some second-order

features) specifically designed for modelling dynamically changing worlds. All changes in the world result from the application of named *actions*, which are denoted by function symbols and may be parameterized. A first-order term called a *situation* represents a sequence of actions or *possible world history*. An *object* is a first-order term used to denote anything that isn't an action or situation. A special constant S_0 indicates the *initial situation* in which no actions have been performed. A distinguished binary function $do(a, s)$ denotes the successor situation resulting from performing action a in situation s. The binary predicate $Poss(a, s)$ is used to indicate that it is possible to perform action a is situation s.

Relations whose truth values can change from situation to situation are referred to as *relational fluents*. Likewise, functions whose mappings can change from situation to situation are referred to as *functional fluents*. A fluent (relational or functional) is denoted by including a situation argument as its last parameter, indicating the fluent's value at that situation. Formulae without situation terms are referred to as *situation independent* formulae and are unchanged by action.

Domain theories are formally defined as follows.

Definition 1 A *basic action theory (BAT)* consists of the following collection of axioms:

1. For each action A, an *action precondition axiom*:
$$Poss(A(\vec{x}), s) \equiv \Pi_A(\vec{x}, s),$$
where Π_A is a first-order formula.[1]
2. For each relational fluent F, a *successor state axiom*:
$$F(\vec{x}, do(a, s)) \equiv \gamma_F^+(\vec{x}, a, s) \vee F(\vec{x}, s) \wedge \neg\gamma_F^-(\vec{x}, a, s),$$
where $\gamma_F^{\pm}$ are first-order formulae (see below).
3. For each functional fluent f, a successor state axiom:
$$f(\vec{x}, do(a, s)) = y \equiv \gamma_f(\vec{x}, y, a, s) \vee f(\vec{x}, s) = y \wedge \neg(\exists y')\gamma_f(\vec{x}, y', a, s),$$
where γ_f is a first-order formulae (see below).
4. A set of *initial situation axioms*, first-order sentences that syntactically only mention the situation term S_0 or no situation term at all.
5. *Unique names axioms* for actions and domain independent *foundational axioms* (see (Reiter 2001)).

(1) characterizes the conditions required for an action A to be applied in s. We will typically ignore *Poss* in this paper and assume that actions are always executable. (2) and (3) provide a solution to the *frame problem*: the value of a fluent at $do(a, s)$ is completely determined by a formula that only mentions situation s. For a relational fluent F, γ_F^+ (similarly, γ_F^-) describes all the ways of making F true (similarly, false) in the situation $do(a, s)$. For a functional fluent f, γ_f characterizes how the mapping of f changes in the situation $do(a, s)$. (4) specifies the initial values of fluents.

[1] Axioms that contain "free" variables can be thought of as being universally quantified from outside the axiom.

A K Fluent in the Situation Calculus

The situation calculus formalism described above does not distinguish between what is *true* in a situation and what is *known* in a situation. Scherl and Levesque (1994; 2003) formalize the idea of *knowledge* in the situation calculus by adapting a possible worlds model of knowledge as was done by Moore (1985). A special binary relation over situations is introduced, $K(s', s)$, read informally as "s' is accessible from s."[2] The K fluent is treated like any other fluent, with the last argument being the official situation argument.

Informally, $K(s', s)$ holds when as far as an agent in situation s knows, it could be in situation s'. Using K we can define the expression **Knows**(ϕ, s) as the abbreviation:

$$\textbf{Knows}(\phi, s) \stackrel{\text{def}}{=} (\forall s').K(s', s) \supset \phi[s'].$$

Here, **Knows**(ϕ, s) means "ϕ is known to be true in situation s." ϕ is any first-order formula and may contain the special situation term *now* meaning "at this moment." The notation $\phi[s]$ indicates that all occurrences of *now* in ϕ should be replaced with s when **Knows**(ϕ, s) is expanded.

Like other fluents, the K fluent requires a successor state axiom that defines how it changes due to action.

Definition 2 A *successor state axiom for* K has the form:

$$K(s'', do(a, s)) \equiv (\exists s').s'' = do(a, s') \wedge K(s', s) \wedge \varphi_1(a, s, s') \wedge \varphi_2(a, s, s') \wedge \ldots \wedge \varphi_m(a, s, s'),$$

where each $\varphi_i(a, s, s')$ is defined to have the form:

$$(\forall \vec{x}_i)\big(a = \alpha_i(\vec{u}_i) \supset \bigwedge\nolimits_{j=1}^{l}(\psi_{ij}(\vec{v}_{ij}, s) \supset (\forall y).\theta_{ij}(\vec{w}_{ij}, y, s) \equiv \theta_{ij}(\vec{w}_{ij}, y, s'))\big).$$

$\theta_{ij}(\vec{w}_{ij}, y, s)$ has the form $F(\vec{w}_{ij}, s)$ for a relational fluent symbol F, or $f(\vec{w}_{ij}, s) = y$ for a functional fluent f.

Each α_i is an action term denoting a *knowledge producing* action. The vectors of variables $\vec{u}_i$, $\vec{v}_{ij}$, and $\vec{w}_{ij}$ must all be (possibly empty) subsets of the variables of $\vec{x}_i$. ψ_{i1} is a situation-independent formula, and each ψ_{ij} $(1 < j \leq l)$ is a situation-independent formula or a formula of the form:

$$\psi_{ij}(\vec{v}_{ij}, s) \stackrel{\text{def}}{=} \psi_{ik}(\vec{v}_{ik}, s) \wedge \big([\neg]\theta_{ik}(\vec{w}_{ik}, y/t, s) \big),$$

where $1 \leq k < j$, and y is syntactically replaced by a ground, situation-independent term t in θ_{ik}.

We will assume that each action is either a knowledge-producing action or an ordinary *physical* action, but not both. Updates to K depend on the type of action that is applied. When a physical action A is applied, the K axiom reduces to the form $K(s'', do(A, s)) \equiv (\exists s').s'' = do(A, s') \wedge K(s', s)$, meaning the size of the set of K-accessible situations is unchanged. Physical actions can cause changes to the values of regular fluents, however, thereby changing the agent's knowledge.

The α_i denote knowledge producing or *sensing* actions that only change the agent's knowledge. (We will typically

[2] We may also say that s' is "K-related" to s.

encode knowledge-producing actions so they leave the values of ordinary fluents unchanged.) Each φ_i ranges over a universally-quantified formula that encodes the effects of a knowledge-producing action α_i, as a conjunction of j "guarded" expressions. Each ψ_{ij} is a *guard condition* for an effect that (conditionally) senses the value of a simple fluent literal expression, θ_{ij}. Guard conditions are expressed in terms of situation s and may not mention other situations. Each fluent mentioned in a guard formula must itself be sensed as an effect of the same action, and the guard conditions of that effect must be included as part of any guard formula that references that fluent. Thus, guard conditions that mention fluents are tightly related to other sensing effects of the same action (see Example 1 below). Applying α_i at s updates K for each guard condition ψ_{ij} that evaluates to "true" so that α_i observes θ_{ij}'s value and the agent comes to know whether θ_{ij} holds or not at $do(\alpha_i, s)$.[3]

Our K axiom differs from the standard K axiom in (Scherl and Levesque 1994) which does not include guard conditions. However, the Scherl and Levesque axiom allows more general expressions to be sensed, since each θ_{ij} can be an arbitrary first-order formula. Our axiom is similar to that of (Petrick and Levesque 2002) and lets us model interesting actions with conditional or universal sensory effects.

Example 1 Consider the K successor state axiom:

$$\begin{aligned} K(s'', do(a,s)) \equiv (\exists s').s'' = do(a,s') \wedge K(s',s) \wedge \\ (\forall x)\ (a = sense_1(x) \supset (F(x,s) \equiv F(x,s'))) \wedge \\ (\forall x)\ (a = sense_2 \supset (F(x,s) \equiv F(x,s'))) \wedge \\ (\forall x)\ (a = sense_3(x) \supset \\ \qquad (f(x,s) = f(x,s') \wedge \\ \qquad (f(x,s) = c \supset (F(x,s) \equiv F(x,s'))). \end{aligned}$$

This axiom encodes the effects of three knowledge-producing actions. $sense_1$ unconditionally senses the truth value of a fluent F for the specified x. $sense_2$ is a more complex action with a *universal* sensory effect: the universal quantification of x results in the unconditional sensing of F for every value of x. Finally, $sense_3$ has a compound effect: it unconditionally senses the value of $f(x)$ and also *conditionally* senses $F(x)$, provided $f(x,s) = c$ is true. Here, $f(x,s) = c$ acts as the guard to the second effect and must evaluate as true before $F(x)$ is sensed. (Definition 2 requires that $f(x)$ be sensed as part of $sense_3$ before it can be used as a guard condition of the same action.) Actions like $sense_3$ allow the properties of a restricted set of objects to be sensed, contingent on the truth of another property. □

The addition of K also requires new foundational axioms (see (Scherl and Levesque 1994)) and axioms that describe the possible world alternatives for K in the initial situation. To this end, $Init(s)$ is defined as the abbreviation:

$$Init(s) \stackrel{\text{def}}{=} \neg(\exists a, s')s = do(a, s').$$

[3]We note that $(\forall y).f(\vec{x},s) = y \equiv f(\vec{x},s') = y$ is logically equivalent to $f(\vec{x},s) = f(\vec{x},s')$ and so a θ_{ij} expression of the form $f(\vec{x},s) = y$ models an effect that senses the referent of f.

$Init(s)$ indicates that "s is an initial situation." Ordinary initial situation axioms are also permitted to specify the values of fluents at any initial situation, not just S_0. Together these axioms describe what is known and not known initially.

Finally, by restricting the accessibility of K, we can also model the properties of particular modal logics of knowledge.[4] To do so, a subset of four axioms defining whether K is reflexive, transitive, Euclidean, and symmetric is included as K's *initial accessibility properties* (see the Appendix). Scherl and Levesque (1994) show that it is sufficient to limit the accessibility of K on initial situations alone, to ensure the same set of properties is preserved through action. We can also establish a similar result for our modified K axiom.

Theorem 1 *If the initial accessibility properties defined for K include a subset of the reflexive, transitive, Euclidean, and symmetric properties, then these same properties hold for all situations, not just initial situations.*

Proof Proof sketches for this theorem, and subsequent results, can be found in the Appendix. ■

We will refer to the above action theories with K as *KBATs*. When describing KBATs we will typically focus on certain axioms (such as the successor state axioms) and assume that all other axioms have been appropriately defined.

KBATs also give rise to a number of basic properties for manipulating various types of **Knows** expressions.

Theorem 2 *Let Σ be a KBAT. If ϕ and ψ are first-order sentences where s appears free in ϕ and ψ, and $\phi(\vec{x}, s)$ is a first-order formula where $\vec{x}$ and s are the only free variables of ϕ, then the following properties hold:*

1. $\Sigma \models (\forall s).\mathbf{Knows}(\phi(now), s) \wedge \mathbf{Knows}(\psi(now), s) \equiv \mathbf{Knows}(\phi(now) \wedge \psi(now), s)$,
2. $\Sigma \models (\forall s).\mathbf{Knows}(\phi(now), s) \vee \mathbf{Knows}(\psi(now), s) \supset \mathbf{Knows}(\phi(now) \vee \psi(now), s)$,
3. $\Sigma \models (\forall s).(\forall \vec{x})\mathbf{Knows}(\phi(\vec{x}, now), s) \equiv \mathbf{Knows}((\forall \vec{x})\phi(\vec{x}, now), s)$.
4. $\Sigma \models (\forall s).(\exists \vec{x})\mathbf{Knows}(\phi(\vec{x}, now), s) \supset \mathbf{Knows}((\exists \vec{x})\phi(\vec{x}, now), s)$.

While the properties in Theorem 2 are not exhaustive, they are nevertheless important since they provide us with a useful set of tools for reasoning about **Knows**. For instance, properties like (1) and (3) provide us with a way of "deconstructing" certain **Knows** formulae into equivalent forms that remove conjunction and universal quantification from the scope of the **Knows** operator. We note that, in general, we cannot perform similar transformations for disjunction and existential quantification, as shown by properties (2) and (4). In the remainder of the paper we will investigate KBATs that give rise to Cartesian situations, and describe a set of additional properties that result from such theories.

[4]See (Fagin et al. 1995) for a good introduction to modal logics of knowledge and their properties.

Cartesian Situations

KBATs may include both relational and functional fluents, however, we will assume that such theories only contain a finite number of fluent symbols. We will also consider simple formulae that characterize primitive assertions about relations and functions. These formulae will act as the building blocks for constructing more complex expressions.

Definition 3 Let Σ be a KBAT. A *primitive fluent literal (pfl)* is a formula of the form: $F(t_1, t_2, \ldots, t_n, s)$, $f(t_1, t_2, \ldots, t_n, s) = t_{n+1}$, $\neg F(t_1, t_2, \ldots, t_n, s)$, or $f(t_1, t_2, \ldots, t_n, s) \neq t_{n+1}$, where F is an ordinary relational fluent symbol, f is a functional fluent symbol, the t_is are situation-free terms, and s is a situation term.

A pfl makes an assertion about the value of a fluent at a situation: a relational fluent is true or false, or a functional fluent is equal or unequal to some mapping. As notation, we will usually denote a pfl as $\theta(\vec{x}, s)$, regardless of the underlying fluent or sign (and sometimes without the situation term). For a pfl that mentions a functional fluent, we will include the function mapping (i.e., t_{n+1} above) as the final argument of $\vec{x}$. We will also use $\overline{\theta}$ to denote the complement of a pfl. It follows from Definition 3 that the complement of a pfl is also a pfl.

Typically, we will consider *collections* of pfls, $\theta_1, \theta_2, \ldots, \theta_n$ ($n \geq 2$), with the property that all arguments (and function mappings) are distinct variables. This will allow us to quantify individually over particular terms, and characterize those instances of a collection that we do not want certain properties to apply to. To do this we will use guard formulae that constrain the arguments of a given collection of pfls.

Definition 4 Let $\theta_1, \theta_2, \ldots, \theta_n$ be a collection of pfls, where the arguments (and function mappings) of the pfls are all distinct variables. Let $\vec{x}_i = \langle x_{i1}, x_{i2}, \ldots, x_{i|\vec{x}_i|} \rangle$ be the non-situation arguments of θ_i, for each $i = 1, 2, \ldots, n$. The formula $\mathit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n)$ is defined as follows:

$$\mathit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n) \stackrel{\text{def}}{=} \bigwedge_{\substack{i=1,\ldots,n \\ j=i+1,\ldots,n}} \Gamma(\theta_i, \theta_j),$$

where $\Gamma(\theta_i, \theta_j)$ is defined by the rules:

1. If the same relational fluent symbol appears in both θ_i and θ_j (regardless of sign) then
$$\Gamma(\theta_i, \theta_j) \stackrel{\text{def}}{=} \begin{cases} x_{i1} \neq x_{j1} \vee \ldots \vee x_{i|\vec{x}_i|} \neq x_{j|\vec{x}_j|} \\ \qquad \text{if } |\vec{x}_i| > 0, \\ \bot \qquad \text{if } |\vec{x}_i| = 0. \end{cases}$$
2. If the same functional fluent symbol appears in both θ_i and θ_j (regardless of sign) then
$$\Gamma(\theta_i, \theta_j) \stackrel{\text{def}}{=} \begin{cases} x_{i1} \neq x_{j1} \vee \ldots \vee x_{i|\vec{x}_i|-1} \neq x_{j|\vec{x}_j|-1} \\ \qquad \text{if } |\vec{x}_i| > 1, \\ \bot \qquad \text{if } |\vec{x}_i| = 1. \end{cases}$$
3. If the fluent symbols that appear in θ_i and θ_j differ then $\Gamma(\theta_i, \theta_j) \stackrel{\text{def}}{=} \top$.

$\mathit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n)$ is a situation-independent formula that will only evaluate as "true" for distinct and non-complementary instances of $\theta_1, \theta_2, \ldots, \theta_n$. The precise definition of $\mathit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n)$ depends on whether or not a fluent symbol appears more than once in a collection of pfls. For instance, $\mathit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n)$ is defined to be $\top$ if no fluent symbol appears multiple times in $\theta_1, \theta_2, \ldots, \theta_n$; $\mathit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n)$ is $\bot$ if a fluent with only a situation argument appears more than once in $\theta_1, \theta_2, \ldots, \theta_n$. Otherwise, $\mathit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n)$ evaluates as "false" for those instantiations of the arguments of $\theta_1, \theta_2, \ldots, \theta_n$ that lead to multiple instances of the same pfl, or an instance of a pfl and its complement, becoming true. For instance, consider the scenario in the following example.

Example 2 If $F(x_1, s)$ and $\neg F(x_2, s)$ are a pair of pfls then $\mathit{DNCI}(F(x_1, s), \neg F(x_2, s)) \stackrel{\text{def}}{=} (x_1 \neq x_2)$ by Definition 4. If we consider a guarded formula:

$$(\forall x_1, x_2, s).\mathit{DNCI}(F(x_1, s), \neg F(x_2, s)) \supset \Phi_F(x_1, x_2, s),$$

where $\Phi_F(x_1, x_2, s)$ has the form $F(x_1, s) \wedge \neg F(x_2, s)$, then $\mathit{DNCI}(F(x_1, s), \neg F(x_2, s))$ guards against the inconsistent instances of $\Phi_F(x_1, x_2, s)$ when $x_1 = x_2$. $\square$

For pfls that mention functional fluents, the (in)equality mapping is not considered. Thus, we can use *DNCI* to guard against instances of pfls that assert multiple mappings for the same function. We will also use Definition 4 to define the formulae arising from a collection of functional fluent terms (which strictly speaking aren't pfls) and will denote such formulae as $\mathit{DNCI}(f_1(\vec{x}_1), f_2(\vec{x}_2), \ldots, f_n(\vec{x}_n))$. In collections of "propositional" pfls—pfls that only have situation arguments—$\mathit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n)$ will always be $\top$ or $\bot$.

We now consider what we mean by a *Cartesian situation*, a structural property that imposes certain conditions on the way a set of K-accessible situations is configured. Our notion of a Cartesian situation is inspired by a Cartesian product over sets, e.g., $\{a, b\} \times \{c, d\} = \{\langle a, c \rangle, \langle b, c \rangle, \langle b, d \rangle, \langle a, d \rangle\}$. In the mathematical sense, a Cartesian product defines a set that contains an ordered vector for every possible combination of elements taken from the original sets. Applying this idea to situations, we would like to ensure that certain configurations of fluent values exist across a set of K-related situations. In particular, if we consider the possible values that individual fluents have at the situations accessible from some situation, we would like to guarantee that there must also exist accessible situations for every possible combination of these fluents' values.

Example 3 Let Σ be a KBAT with two relational fluents, F and G, and let t be a ground situation term such that

$$\Sigma \models (\exists s_1, s_2, s_3, s_4).K(s_1, t) \wedge K(s_2, t) \wedge K(s_3, t) \wedge K(s_4, t) \wedge F(s_1) \wedge G(s_2) \wedge \neg F(s_3) \wedge \neg G(s_4).$$

In this case, we have situations K-related to t where F and G each individually take on positive and negative values. If t is a Cartesian situation then we would like to ensure that there exist situations K-related to t for each combination of the possible values of F and G. Thus, we would expect that

$$\Sigma \models (\exists s_1, s_2, s_3, s_4).K(s_1, t) \wedge K(s_2, t) \wedge K(s_3, t) \wedge K(s_4, t) \wedge F(s_1) \wedge G(s_1) \wedge \neg F(s_2) \wedge G(s_2) \wedge F(s_3) \wedge \neg G(s_3) \wedge \neg F(s_4) \wedge \neg G(s_4).$$

We note that the necessity of certain fluent configurations is based solely on the values of individual fluents at other accessible situations: if $\neg G$ did not hold at any situation accessible from t then we would not require a situation where F and $\neg G$ held, or where $\neg F$ and $\neg G$ held. $\square$

Another observation is that the elements of a mathematical Cartesian product set are all interrelated. In particular, each element has the property that it only differs from some other element by one component. If we extend this idea to situations, then the Cartesian property must ensure that individual situations are configured to be identical to other situations, except for certain "minimal" differences. It is this characterization that we will use as the motivation for our formal definition of a Cartesian situation.

Definition 5 Let Σ be a KBAT with a finite number of fluent symbols. Let $F_1, F_2, \ldots, F_n$ be the relational fluent symbols of Σ, and let $f_1, f_2, \ldots, f_m$ be the functional fluent symbols of Σ. The abbreviation *Cart*(s) is defined as the expression:

$$\begin{aligned}
&\big[\textstyle\bigwedge_{i=1}^{n}(\forall s', s'', \vec{x}_i).K(s', s) \wedge K(s'', s) \wedge \\
&\quad F_i(\vec{x}_i, s') \not\equiv F_i(\vec{x}_i, s'') \supset \\
&\qquad (\exists s^*).K(s^*, s) \wedge F_i(\vec{x}_i, s^*) \equiv F_i(\vec{x}_i, s'') \wedge \\
&\qquad (\forall \vec{x}_j)(\vec{x}_j \neq \vec{x}_i \supset F_i(\vec{x}_j, s^*) \equiv F_i(\vec{x}_j, s')) \wedge \\
&\qquad \textstyle\bigwedge_{\substack{k=1 \\ k\neq i}}^{n}(\forall \vec{x}_k) F_k(\vec{x}_k, s^*) \equiv F_k(\vec{x}_k, s') \wedge \\
&\qquad \textstyle\bigwedge_{k=1}^{m}(\forall \vec{x}_k) f_k(\vec{x}_k, s^*) = f_k(\vec{x}_k, s')\big] \wedge \\
&\big[\textstyle\bigwedge_{i=1}^{m}(\forall s', s'', \vec{x}_i).K(s', s) \wedge K(s'', s) \wedge \\
&\quad f_i(\vec{x}_i, s') \neq f_i(\vec{x}_i, s'') \supset \\
&\qquad (\exists s^*).K(s^*, s) \wedge f_i(\vec{x}_i, s^*) = f_i(\vec{x}_i, s'') \wedge \\
&\qquad (\forall \vec{x}_j)(\vec{x}_j \neq \vec{x}_i \supset f_i(\vec{x}_j, s^*) = f_i(\vec{x}_j, s')) \wedge \\
&\qquad \textstyle\bigwedge_{\substack{k=1 \\ k\neq i}}^{m}(\forall \vec{x}_k) f_k(\vec{x}_k, s^*) = f_k(\vec{x}_k, s') \wedge \\
&\qquad \textstyle\bigwedge_{k=1}^{n}(\forall \vec{x}_k) F_k(\vec{x}_k, s^*) \equiv F_k(\vec{x}_k, s')\big].
\end{aligned}$$

Intuitively, *Cart*(s) means that "situation s is a *Cartesian situation*." Our definition of *Cart*(s) is based on identifying pairs of situations that differ on the value of some instantiated fluent. Such a pair implies the existence of an accessible situation that is a cross between the two situations: the "differing" fluent value from one situation is true at this new situation, while the values of all other fluents are otherwise identical to those of the second situation. Although we only consider theories with a finite number of fluent symbols, we do not restrict how these symbols may be instantiated.

Using Definition 5, we can also demonstrate that our notion of a Cartesian situation satisfies our earlier intuition concerning the existence of certain configurations of fluent values. We revisit the scenario from Example 3.

Example 4 Consider the KBAT Σ in Example 3 where F and G are relational fluents, t is a ground situation term, and

$$\begin{aligned}
\Sigma \models (\exists s_1, s_2, s_3, s_4).&K(s_1, t) \wedge K(s_2, t) \wedge K(s_3, t) \wedge \\
&K(s_4, t) \wedge F(s_1) \wedge G(s_2) \wedge \neg F(s_3) \wedge \neg G(s_4).
\end{aligned}$$

Say $\Sigma \models$ *Cart*$(t) \wedge (\exists s_1).K(s_1, t) \wedge F(s_1) \wedge \neg G(s_1)$ (we will have similar conclusions if $G(s_1)$ holds). Since $\Sigma \models (\exists s_2).K(s_2, t) \wedge G(s_2)$, it then follows from Definition 5 that $\Sigma \models (\exists s_2').K(s_2', t) \wedge F(s_2') \wedge G(s_2')$. Also, since $\Sigma \models (\exists s_3).K(s_3, t) \wedge \neg F(s_3)$ we have that $\Sigma \models (\exists s_3').K(s_3', t) \wedge \neg F(s_3') \wedge G(s_3')$. Finally, since $\Sigma \models (\exists s_1).K(s_1, t) \wedge \neg G(s_1)$ it also follows that $\Sigma \models (\exists s_4').K(s_4', t) \wedge \neg F(s_4') \wedge \neg G(s_4')$. Thus, there exist situations K-related to t for each combination of the possible values of F and G.

Say instead $\Sigma \models (\exists s_1, s_2).K(s_1, t) \wedge K(s_2, t) \wedge F(s_1) \wedge \neg G(s_1) \wedge G(s_2)$ but $\Sigma \not\models (\exists s').K(s', t) \wedge F(s') \wedge G(s')$. Thus, there is no K-related situation where both F and G hold, and it follows from Definition 5 that $\Sigma \not\models$ *Cart*(t). $\square$

The accessibility of K also plays a role in determining if a situation is Cartesian or not. Recall that K may be initially constrained by a subset of four accessibility properties, and that Theorem 1 guarantees that any subset of these properties also holds for all situations. If certain combinations of these properties are known to hold, then it follows that some sets of situations must necessarily be "mutually" Cartesian.

Theorem 3 *Let Σ be a KBAT. If the initial accessibility properties of K satisfy the conditions that K is (i) symmetric and Euclidean, (ii) reflexive and Euclidean, (iii) symmetric and transitive, or (iv) transitive and Euclidean, then* $\Sigma \models (\forall s_1, s_2).\big($*Cart*$(s_1) \wedge K(s_2, s_1)\big) \supset$ *Cart*(s_2).

Properties of Cartesian Situations

The structural restrictions of Cartesian situations give rise to a set of properties that can be used to decompose complex formulae with **Knows** into simpler but equivalent components, letting us reason about a formula's constituent parts. Although the standard properties of **Knows** already let us do this to some extent (e.g., see Theorem 2), Cartesian properties are much richer. They also help us to better understand the restrictions we must accept with Cartesian situations.

We begin by formalizing our earlier intuition about Cartesian situations, that certain configurations of fluent values must occur across a set of K-related situations.

Theorem 4 *Let Σ be a KBAT. If $\theta_1, \theta_2, \ldots, \theta_n$ ($n \geq 2$) is a collection of pfls, where the arguments of each θ_i are all variables, denoted by $\vec{x}_i$, and each variable of $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ appears no more than once, then*

$$\begin{aligned}
\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n).&\textit{Cart}(s) \wedge \textit{DNCI}(\theta_1, \theta_2, \ldots, \theta_n) \supset \\
&\big(\textstyle\bigwedge_{i=1}^{n}(\exists s_i)(K(s_i, s) \wedge \theta_i[s_i])\big) \supset \\
&\qquad (\exists s^*)(K(s^*, s) \wedge \textstyle\bigwedge_{i=1}^{n}\theta_i[s^*]).
\end{aligned}$$

Theorem 4 illustrates a strong structural property about Cartesian situations, ensuring the existence of particular combinations of fluent values, as in Example 3 and Example 4. (This idea is analogous to the existence of particular vectors in standard Cartesian product sets.) Note that this result is guarded by *Cart* and *DNCI*, as other results will be.

We can now establish a set of properties for Cartesian situations that let us decompose many common types of epistemic formulae into simpler components mentioning pfls.

Theorem 5 *Let Σ be a KBAT. If $\theta_1, \theta_2, \ldots, \theta_n$ ($n \geq 2$) is a collection of pfls, where the arguments of each θ_i are all variables, denoted by $\vec{x}_i$, and each variable of $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ appears no more than once; and $f_1(\vec{x}_1), f_2(\vec{x}_2), \ldots, f_n(\vec{x}_n)$ ($n \geq 2$) are terms, where each f_i is a functional fluent symbol, each $\vec{x}_i$ is a vector of variables with no variable appearing more than once in $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$, and y is a variable that doesn't appear in $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$; then the following properties hold:*

1. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n).Cart(s) \wedge DNCI(\theta_1, \theta_2, \ldots, \theta_n) \supset$
$\big[\mathbf{Knows}(\bigvee_{i=1}^{n} \theta_i(now), s) \equiv \bigvee_{i=1}^{n} \mathbf{Knows}(\theta_i(now), s)\big]$
2. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2).Cart(s) \wedge DNCI(\theta_1, \theta_2) \supset$
$\big[\mathbf{Knows}(\theta_1(now) \supset \theta_2(now), s) \equiv$
$(\mathbf{Knows}(\overline{\theta}_1(now), s) \vee \mathbf{Knows}(\theta_2(now), s))\big]$
3. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n).Cart(s) \wedge DNCI(\theta_1, \theta_2, \ldots, \theta_n) \supset$
$\big[\mathbf{Knows}(\bigwedge_{i=1}^{n-1} \theta_i(now) \equiv \theta_{i+1}(now), s) \equiv$
$(\bigwedge_{i=1}^{n} \mathbf{Knows}(\theta_i(now), s) \vee \bigwedge_{i=1}^{n} \mathbf{Knows}(\overline{\theta}_i(now), s))\big]$
4. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n).Cart(s) \wedge$
$DNCI(f_1(\vec{x}_1), f_2(\vec{x}_2), \ldots, f_n(\vec{x}_n)) \supset$
$\big[\mathbf{Knows}(\bigwedge_{i=1}^{n-1} f_i(\vec{x}_i, now) = f_{i+1}(\vec{x}_{i+1}, now), s) \equiv$
$(\exists y).\bigwedge_{i=1}^{n} \mathbf{Knows}(f_i(\vec{x}_i, now) = y, s)\big]$
5. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n).Cart(s) \wedge$
$DNCI(f_1(\vec{x}_1), f_2(\vec{x}_2), \ldots, f_n(\vec{x}_n)) \supset$
$\big[\mathbf{Knows}(\bigvee_{i=1}^{n-1} f_i(\vec{x}_i, now) \neq f_{i+1}(\vec{x}_{i+1}, now), s) \equiv$
$(\forall y)\bigvee_{i=1}^{n} \mathbf{Knows}(f_i(\vec{x}_i, now) \neq y, s)\big]$.

Theorem 5 illustrates that when situations are Cartesian, certain types of knowledge (i.e., certain configurations of K-accessible situations) are prohibited from occurring, giving rise to a set of knowledge decomposition properties. The list of properties in Theorem 5 is not exhaustive, however, it does provide us with a set of tools for simplifying many types of epistemic formulae involving **Knows**.

(1) establishes a fundamental result for Cartesian situations: it allows us to decompose knowledge of a disjunctive formula into an equivalent form where we can reason about knowledge of the individual disjuncts. This property comes at a representational cost, namely that we require definite knowledge of some of the component parts that make up the disjunction. (1) does not prohibit disjunction entirely, but instead has the effect of removing any inter-dependencies among the primitive components (i.e., fluents). Thus, certain "general" disjunctions cannot be represented in theories with Cartesian restrictions. In practical terms, this means that such theories will be inherently less expressive than theories without such restrictions. For example, we can no longer explicitly represent knowledge of $F \vee G$ without definite knowledge of F or definite knowledge of G.

(2) illustrates that knowledge of a logical implication reduces to a disjunction of **Knows** expressions. Applying this result to logical equivalence produces the disjunctive normal form in (3). In particular, if a set of pfls are known to be equivalent then either each pfl must be known to be true, or the complement of each pfl must be known to be true. Thus, we "know whether" each formula is true or not and coming to know the truth of one formula means we also come to know the truth of the others.

(4) and (5) apply to knowledge of functional equality and inequality, respectively. For (4) we note that the formula $\mathbf{Knows}(\bigwedge_{i=1}^{n-1} f_i(now) = f_{i+1}(now), s)$ is logically equivalent to $\mathbf{Knows}((\exists y).\bigwedge_{i=1}^{n} f_i(now) = y, s)$, and so the consequent of (4) lets us remove an existential quantifier from the context of the **Knows** operator. This relationship indicates a situation where *de dicto* and *de re* knowledge are equivalent. (This equivalence doesn't hold in general, nor is it desirable). (5) establishes the dual of (4) and ensures that when functions are unequal they share no common mappings across the space of K-related situations (the set of possible mappings across all accessible situations is disjoint).[5] We illustrate this latter property in the following example.

Example 5 Consider a KBAT Σ where f_1 and f_2 are functional fluent symbols and c_1 and c_2 are distinct constants. Let t be a ground situation term and say that

$$\Sigma \models \mathbf{Knows}(f_1(now) \neq f_2(now), t), \quad (1)$$
$$\Sigma \models \neg\mathbf{Knows}(f_2(now) \neq c_1, t), \quad (2)$$
$$\Sigma \models \mathbf{Knows}(f_1(now) = c_1 \vee f_1(now) = c_2, t). \quad (3)$$

Say $\Sigma \models Cart(t)$. Then, from (1) and Theorem 5(5) we have that $\Sigma \models (\forall y).\mathbf{Knows}(f_1(now) \neq y, t) \vee \mathbf{Knows}(f_2(now) \neq y, t)$ and so $\Sigma \models \mathbf{Knows}(f_1(now) \neq c_1, t) \vee \mathbf{Knows}(f_2(now) \neq c_1, t)$. Since (2) holds, it also follows that $\Sigma \models \mathbf{Knows}(f_1(now) \neq c_1, t)$, and from the ordinary properties of **Knows** we can conclude that $\Sigma \models \mathbf{Knows}(f_1(now) = c_2, t)$. (If t was not Cartesian then this conclusion would not necessarily follow.) Similarly, from (1) and Theorem 5(5) we also have that $\Sigma \models \mathbf{Knows}(f_1(now) \neq c_2, t) \vee \mathbf{Knows}(f_2(now) \neq c_2, t)$. Since $\Sigma \models \mathbf{Knows}(f_1(now) = c_2, t)$ it also follows from standard **Knows** reasoning that $\Sigma \models \mathbf{Knows}(f_2(now) \neq c_2, t)$. □

As we have seen, Cartesian situations provide us with new properties that augment those in Theorem 2 for simplifying **Knows** formulae. In practice, however, we may first have to detect and reason away certain "hidden" logical constraints (such as tautologies), before a formula can be expressed in a form that permits further simplifications to be made. In general this can be a difficult task, but we consider how this can be done for one particular class of epistemic formulae.

Theorem 6 *Let Σ be a KBAT. Let ϕ be a ground, quantifier-free formula without equality that doesn't mention functional fluents or K, and only mentions the situation term s, where s is free. If $\Pi(\phi) = \{\pi_1, \pi_2, \ldots, \pi_n\}$ is the set of prime implicates of ϕ, where each π_i has the form $(\alpha_{i1} \vee \alpha_{i2} \vee \ldots \vee \alpha_{i|\pi_i|})$, then*

$$\Sigma \models (\forall s).Cart(s) \supset$$
$$\big[\mathbf{Knows}(\phi(now), s) \equiv \bigwedge_{i=1}^{n} \big(\bigvee_{j=1}^{|\pi_i|} \mathbf{Knows}(\alpha_{ij}(now), s)\big)\big].$$

[5] E.g., for two functions f_1 and f_2, (5) is equivalent to the world-level property $(\forall s', s'').K(s', s) \wedge K(s'', s) \supset f_1(s') \neq f_2(s'')$, when f_1 and f_2 are known to be unequal at a Cartesian situation s.

Theorem 6 identifies a particular class of epistemic formulae that can be decomposed as completely as possible, into **Knows** components that only mention fluent literals. This is done by generating a set of prime implicates. We do not focus here on methods for finding sets of prime implicates, which can be a computationally expensive process.

Progressing Cartesian Situations

In general, Cartesian situations are "brittle" and even simple actions can often break the Cartesian structure when applied. However, we can also identify expressive classes of KBATs that preserve the Cartesian property through action. We will investigate some of these action theories by considering knowledge producing and physical actions separately.

For knowledge producing actions we need only consider the form of the K successor state axiom, since such actions only change K and not ordinary fluents. It turns out that the form of the K axiom we give in Definition 2 ensures that if a situation is Cartesian then applying a sensing action preserves the Cartesian property. We have the following result.

Theorem 7 *Let Σ be a KBAT. If α is a ground action term denoting a knowledge-producing action then* $\Sigma \models (\forall s).\mathit{Cart}(s) \supset \mathit{Cart}(\mathit{do}(\alpha, s))$.

This result is important since it means that sensing actions as we've defined them will not break the Cartesian property. This is a direct consequence of the form of the K axiom, which models actions that sense instances of fluent literals.

If we consider more expressive knowledge-producing actions than those permitted by Definition 2, then it becomes easy to construct theories that do not preserve the Cartesian property through action. For instance, in the next example we consider a more general K successor state axiom that breaks the Cartesian property.

Example 6 Let Σ be a theory defined by the axioms:

$$F(\mathit{do}(a, s)) \equiv F(s),$$
$$G(\mathit{do}(a, s)) \equiv G(s),$$
$$K(s'', \mathit{do}(a, s)) \equiv (\exists s').s'' = \mathit{do}(a, s') \wedge K(s', s) \wedge$$
$$(a = \mathit{sense} \supset ((F(s') \wedge G(s')) \equiv (F(s) \wedge G(s)))),$$
$$(\exists s_1, s_2, s_3, s_4).K(s_1, S_0) \wedge K(s_2, S_0) \wedge K(s_3, S_0) \wedge$$
$$K(s_4, S_0) \wedge F(s_1) \wedge G(s_1) \wedge F(s_2) \wedge \neg G(s_2) \wedge$$
$$\neg F(s_3) \wedge G(s_3) \wedge \neg F(s_4) \wedge \neg G(s_4),$$
$$F(S_0), \quad \neg G(S_0).$$

Here *sense* is a knowledge-producing action that senses the truth of a simple conjunctive formula, $F \wedge G$. Initially nothing is known about F or G and $\Sigma \models \mathit{Cart}(S_0)$. Applying *sense* produces situations K-related to $\mathit{do}(\mathit{sense}, S_0)$ for every configuration of F and G, except where both F and G hold. Thus, $\Sigma \not\models \mathit{Cart}(\mathit{do}(\mathit{sense}, S_0))$. (We note that **Knows**$(\neg F(\mathit{now}) \vee \neg G(\mathit{now}), \mathit{do}(\mathit{sense}, S_0))$ holds but neither **Knows**$(\neg F(\mathit{now}), \mathit{do}(\mathit{sense}, S_0))$ nor **Knows**$(\neg G(\mathit{now}), \mathit{do}(\mathit{sense}, S_0))$ hold.) □

For ordinary physical actions, it is also easy to construct KBATs whose actions do not preserve the Cartesian property when applied. For instance, consider the following example.

Example 7 Let Σ be a KBAT defined by the axioms:

$$F(\mathit{do}(a, s)) \equiv (a = \mathit{change} \wedge G(s)) \vee F(s),$$
$$G(\mathit{do}(a, s)) \equiv G(s),$$
$$K(s'', \mathit{do}(a, s)) \equiv (\exists s').s'' = \mathit{do}(a, s') \wedge K(s', s),$$
$$(\exists s_1, s_2, s_3, s_4).K(s_1, S_0) \wedge K(s_2, S_0) \wedge K(s_3, S_0) \wedge$$
$$K(s_4, S_0) \wedge F(s_1) \wedge G(s_1) \wedge F(s_2) \wedge \neg G(s_2) \wedge$$
$$\neg F(s_3) \wedge G(s_3) \wedge \neg F(s_4) \wedge \neg G(s_4),$$
$$F(S_0), \quad G(S_0).$$

F and G are initially unknown and $\Sigma \models \mathit{Cart}(S_0)$. Applying *change* produces situations K-related to $\mathit{do}(\mathit{change}, S_0)$ for every configuration of F and G, except where both $\neg F$ and G hold. Thus, $\Sigma \not\models \mathit{Cart}(\mathit{do}(\mathit{change}, S_0))$. (We note that **Knows**$(F(\mathit{now}) \vee \neg G(\mathit{now}), \mathit{do}(\mathit{change}, S_0))$ holds but neither **Knows**$(F(\mathit{now}), \mathit{do}(\mathit{change}, S_0))$ nor **Knows**$(\neg G(\mathit{now}), \mathit{do}(\mathit{change}, S_0))$ hold.) □

Since physical actions affect the ordinary fluents of a KBAT, we will consider certain classes of KBATs formed by restricting the ordinary successor state axioms.

Context-Free Theories The first class of KBATs we consider are formed by restricting the $\gamma_F^{\pm}$ and γ_f components of successor state axioms in Definition 1 to be situation independent formulae (i.e., no fluents can be mentioned). Such axioms are referred to as *context-free successor state axioms* (Lin and Reiter 1997). Quantification is still permitted, provided the quantifiers only range over the situation-independent formulae. If every successor state axiom is context free then a KBAT is said to be a *context-free theory*. We have the following result for context-free KBATs.

Theorem 8 *If Σ is a context-free theory then it follows that* $\Sigma \models (\forall s)(\mathit{Init}(s) \supset \mathit{Cart}(s)) \supset (\forall s)\mathit{Cart}(s)$.

Since situation-independent formulae are evaluated independent of individual situations, updates to fluents resulting from context-free successor state axioms are made uniformly across a set of K-related situations. Thus, if all initial situations are Cartesian then all successor situations will remain Cartesian. Although context-free theories limit fluent references, these types of axioms are important and arise quite often in practice. For instance, STRIPS-style planning actions (Fikes and Nilsson 1971)—actions with unconditional effects—are a subset of this class of actions.

Fluent Dependent Theories We next consider a class of KBATs that allows actions to have particular conditional effects that depend on the value of some fluent. These theories are inspired by successor state axioms of the form:

$$F(\mathit{do}(a, s)) \equiv (a = A \wedge G(s)) \vee F(s) \wedge \neg(a = A \wedge \neg G(s))$$
$$G(\mathit{do}(a, s)) \equiv a = A \vee G(s).$$

In this case, when A is applied the truth of F in a successor situation depends on the current truth value of G; A also causes G to become true in a successor situation. If we instead had the following successor state axiom for G:

$$G(\mathit{do}(a, s)) \equiv (a = A \wedge F(s)) \vee G(s) \wedge \neg(a = A \wedge \neg F(s))$$

then the value of the fluent F at a successor situation would depend on G, and the value of G would depend on F. If we

look at the changes across a set of K-related situations we see that they amount to a permutation or "re-mapping" of the fluent values across the set. New values, when introduced, are made in a uniform manner to all situations.

In general, we consider theories that allow particular "chains" (similar to the first example) and "cycles" (similar to the second example) of dependencies among fluents.

Definition 6 A *fluent dependent theory* restricts the form of the ordinary successor state axioms as follows.

1. For each physical action β, define an associated vector of variables $\vec{y}_\beta = \langle y_1, y_2, \ldots, y_{|\beta|} \rangle$ that will always appear as the non-situation arguments to β whenever β appears.
2. A successor state axiom for each ordinary relational fluent F has the general form in Definition 1, where $\gamma_F^+(\vec{x}, a, s)$ and $\gamma_F^-(\vec{x}, a, s)$ each have the form:

$$\phi(\vec{x}, a) \vee \bigvee_{i=1}^{n} \psi_i(\vec{x}, a, s).$$

ϕ is any situation independent formula whose free variables are among those in $\vec{x}$ and a. Each ψ_i has the form:

$$\psi_i(\vec{x}, a, s) \stackrel{\text{def}}{=} (\exists \vec{w}).a = \beta(\vec{y}_\beta) \left\{ \wedge L(\vec{z}, s) \right\},$$

where β denotes a physical action with associated parameters $\vec{y}_\beta$; the variables of $\vec{x}$ and $\vec{z}$ must be variables of $\vec{y}_\beta$; the variables of $\vec{w}$ are the variables of $\vec{y}_\beta$ not mentioned in $\vec{x}$; and (optionally) L is a relational fluent literal. Also,

(a) For each ψ_i in γ_F^+ (similarly, γ_F^-) that mentions a fluent literal L, some ψ_j in γ_F^- (similarly, γ_F^+) must be syntactically identical to ψ_i with the exception that L is replaced with the complement of L.

(b) Each action name β can only be mentioned at most once in γ_F^+ or γ_F^-.

(c) If ψ_i mentions an action β and a fluent symbol G with arguments $\vec{z}$, then the successor state axiom for G must have the general syntactic form in Definition 1, and the action β must be mentioned in γ_G^+ or γ_G^- (subject to the above requirements of a successor state axiom).

(We have a similar definition for functional fluents.)

The strong syntactic restrictions in Definition 6 have important consequences for Cartesian situations. Intuitively, they ensure that actions produce "symmetric" behaviour across sets of K-related situations. As a result, each action "shuffles" certain components (i.e., fluent values) at each situation, while retaining the overall Cartesian structure.

Theorem 9 *If Σ is a fluent dependent theory then it follows that $\Sigma \models (\forall s)(\mathit{Init}(s) \supset \mathit{Cart}(s)) \supset (\forall s)\mathit{Cart}(s)$.*

Propositional Theories We close this section by considering *propositional KBATs (pKBATs)*, action theories without functional fluents where each ordinary relational fluent has the property that its only argument is the situation argument. Such theories are restrictive but important. For instance, theories with finite domains of discourse can be converted to this form. Many planning algorithms also use techniques that first "propositionalize" planning domains. By focusing on pKBATs, our definitions of pfls, *DNCI*, and Cartesian situations can be simplified due to the restrictions on functional fluents and fluent arguments. As a result, the requirements for Cartesian situations are also much simpler.

Theorem 10 *Let Σ be a pKBAT and let t be a ground situation term. $\Sigma \models \mathit{Cart}(t)$ iff for every collection of pfls, $\theta_1, \theta_2, \ldots, \theta_n$, where no fluent appears more than once,*

$$\Sigma \models \mathbf{Knows}(\bigvee_{i=1}^{n} \theta_i(\mathit{now}), t) \supset \bigvee_{i=1}^{n} \mathbf{Knows}(\theta_i(\mathit{now}), t).$$

Theorem 10 provides an alternate definition for Cartesian situations, expressed at the "knowledge level" rather than the level of K-related situations. Thus, to determine if a situation is Cartesian it suffices to ensure that particular **Knows** formulae can be decomposed into their component parts. We also establish a related result for certain action sequences.

Theorem 11 *Let Σ be a pKBAT. Let t be a ground situation term and let $\vec{A}$ be any finite sequence (possibly empty) of ground action terms denoting physical actions. $\Sigma \models \mathit{Cart}(do(\vec{A}, t))$ iff for every collection of pfls, $\theta_1, \theta_2, \ldots, \theta_n$, where no fluent symbol appears more than once,*

$$\Sigma \models \mathbf{Knows}(\bigvee_{i=1}^{n} \theta_i(do(\vec{A}, \mathit{now})), t) \supset \bigvee_{i=1}^{n} \mathbf{Knows}(\theta_i(do(\vec{A}, \mathit{now})), t).$$

Theorem 11 differs from Theorem 10 by placing the knowledge requirements for Cartesian situations in terms of a predecessor situation, rather than the Cartesian situation itself. Thus, provided no action sequence gives rise to particular epistemic disjunctions that can't be decomposed, the resulting situation will be Cartesian. What Theorem 11 does *not* provide is an efficient method for detecting all the conditions that must hold to ensure a situation is Cartesian. As a result, we may potentially have to consider disjunctions that arise from any sequence of physical actions.

Examples

Although the KBATs in the previous section limit the form of their successor state axioms, they still let us model some interesting actions and domains. We now consider some examples of such theories. In each case, we only highlight the successor state axioms and simply assume that other axioms are appropriately defined. (E.g., initial situation axioms must be properly defined to ensure that $\mathit{Cart}(S_0)$ holds.)

Example 8 In this example we consider a domain inspired by a UNIX-style environment, with three ordinary fluents and three actions. The fluent $\mathit{indir}(f, d, s)$ indicates that "file f is in directory d in situation s." The fluent $\mathit{readable}(f, s)$ denotes that "file f is readable in situation s." A functional fluent $\mathit{size}(f, s) = y$ indicates that "file f has size y in situation s." The action $\mathit{chmod}\text{+}r(f)$ is a physical action that makes a file f readable. Similarly, $\mathit{chmod}\text{-}r(f)$ makes f unreadable. The action $ls(d)$ is a knowledge-producing action that provides information about the files in directory d. The successor state axioms are given as follows:

$$\mathit{indir}(f, d, do(a, s)) \equiv \mathit{indir}(f, d, s),$$
$$\mathit{size}(f, do(a, s)) = y \equiv \mathit{size}(f, s) = y,$$
$$\mathit{readable}(f, do(a, s)) \equiv a = \mathit{chmod}\text{+}r(f) \vee \mathit{readable}(f, s) \wedge \neg(a = \mathit{chmod}\text{-}r(f)),$$

$$K(s'', do(a,s)) \equiv (\exists s').s'' = do(a,s') \wedge K(s',s) \wedge$$
$$(\forall d).(a = ls(d) \supset$$
$$(\forall f)\ (indir(f,d,s) \equiv indir(f,d,s')) \wedge$$
$$(\forall f)\ (indir(f,d,s) \supset (size(f,s) = size(f,s'))) \wedge$$
$$(\forall f)\ (indir(f,d,s) \supset$$
$$(readable(f,s) \equiv readable(f,s')))).$$

The ordinary successor state axioms meet the requirements of a context-free theory. In this case, *indir* and *size* are unchanged by action. The successor state axiom for *readable* encodes the (unconditional) effects of the *chmod+r* and *chmod-r* actions on *readable*: *chmod+r*(f) makes f readable, while *chmod-r*(f) makes f unreadable.

The K axiom has the form required by Definition 2 and describes two types of sensory effects for *ls*. First, it encodes a universal effect: *ls* senses the files f that are in directory d (i.e., all f that satisfy $indir(f,d,s)$). Second, it encodes a conditional sensing effect: besides sensing the contents of directory d, *ls* also senses the size and readability of the files in d (i.e., the truth of $readable(f,s)$ and the denotation of $size(f,s)$ for all f such that $indir(f,d,s)$ is true). □

Example 9 In this example we consider two switches, each of which can be turned on or off. We include two fluents, on_1 and on_2, that represent the state of each switch. We also have two actions, $flip_1$ and $flip_2$, each of which toggles the state of a particular switch. The effects of these actions are encoded in the following pair of successor state axioms:

$$on_1(do(a,s)) \equiv (a = flip_1 \wedge \neg on_1(s)) \vee$$
$$on_1(s) \wedge \neg(a = flip_1 \wedge on_1(s)),$$
$$on_2(do(a,s)) \equiv (a = flip_2 \wedge \neg on_2(s)) \vee$$
$$on_2(s) \wedge \neg(a = flip_2 \wedge on_2(s)).$$

These axioms satisfy our definition of a fluent dependent theory. In particular, each axiom encodes a simple one-fluent "cyclic" dependency: flipping a switch that is off turns it on; flipping it again turns it off.

The axioms also meet the requirements of a propositional KBAT, which lets us use results like Theorem 10 to simplify the definition of a Cartesian situation.[6] For instance, determining if S_0 is Cartesian reduces to verifying that

$$\mathbf{Knows}(\theta_1(now) \vee \theta_2(now), S_0) \supset$$
$$(\mathbf{Knows}(\theta_1(now), S_0) \vee \mathbf{Knows}(\theta_2(now), S_0))$$

holds for each valid pair of pfls, θ_1 and θ_2. (With two fluents we need only consider four pairs of pfls: $on_1(s)$ and $on_2(s)$, $on_1(s)$ and $\neg on_2(s)$, $\neg on_1(s)$ and $on_2(s)$, and $\neg on_1(s)$ and $\neg on_2(s)$.) □

Example 10 In this example, we consider the representation of a simple data buffer that can hold three units of data. Data is written to the buffer by the action $write(d)$. Three functional fluents, B_1, B_2, and B_3, represent the data cells of the buffer. We have the following successor state axioms:

$$B_1(do(a,s)) = y \equiv (\exists d)(a = write(d) \wedge y = d) \vee$$
$$B_1(s) = y \wedge \neg(\exists y')((\exists d).a = write(d) \wedge y' = d),$$
$$B_2(do(a,s)) = y \equiv ((\exists d).a = write(d) \wedge B_1(s) = y) \vee$$
$$B_2(s) = y \wedge \neg(\exists y')((\exists d).a = write(d) \wedge B_1(s) = y'),$$
$$B_3(do(a,s)) = y \equiv ((\exists d).a = write(d) \wedge B_2(s) = y) \vee$$
$$B_3(s) = y \wedge \neg(\exists y')((\exists d).a = write(d) \wedge B_2(s) = y').$$

Here, new data is inserted at one end of the buffer (B_1) and existing data in the buffer is shifted when a new data item is written. (I.e., B_2 takes B_1's value and B_3 takes B_2's value.) Subsequent *write* actions cause further shifts, with the existing data being overwritten. These axioms satisfy the requirements of a "linear" fluent dependent theory. □

Discussion

Cartesian situations are important since they give rise to action theories that let us decompose epistemic formulae into more primitive expressions of knowledge based on fluent literals. These properties come at a representational cost, since they limit the expressiveness of certain types of knowledge (most notably disjunctive knowledge). This trade-off also offers some computational advantages, however. For instance, by focusing on theories that reduce the general problem of reasoning about knowledge to the problem of reasoning about knowledge of fluent literals, we can look to alternative representations that limit the types of knowledge they can represent in similar ways (and potentially avoid the computational drawbacks of possible worlds reasoning). This work offers the possibility of identifying candidate theories that can be compiled into such alternative representations. (The precise gain in terms of computational efficiency will depend on the target representation.) As a result, this work is also closely related to approaches that model knowledge and action without possible worlds and an accessibility relation.

For instance, Petrick and Levesque (2002) transform standard Scherl and Levesque theories into an equivalent form based on *knowledge fluents* (Demolombe and Pozos Parra 2000)—relational fluents that explicitly represent the agent's knowledge of ordinary fluents that are known to be true and known to be false. In this case, determining whether n atomic formulae are known reduces from reasoning about potentially 2^n possible worlds (in the worst case) to reasoning about the truth of $3n$ fluents. Furthermore, knowledge change is treated as ordinary fluent change without the need for a K fluent. This approach relies on theories that limit disjunctive knowledge in a similar way to Cartesian situations. Soutchanski (2001) explores a similar approach, but focuses on less expressive theories than we consider here.

Liu and Levesque (2005) investigate the projection problem for a generalized database called a *proper KB*. Proper KBs do not permit general disjunctions and, thus, appear to be closely related to Cartesian-restricted theories. Liu and Levesque use a simpler notion of sensing than the model presented here, however, and only work with relational flu-

[6] Alternatively, we could model this domain with the successor state axiom $on(x, do(a,s)) \equiv a = flip(x) \wedge \neg on(x,s) \vee on(x,s) \wedge \neg(a = flip(x) \wedge on(x,s))$ and use constants like sw_1 and sw_2 to denote switches. This axiom still has the form required for a fluent dependent theory, but not a propositional theory.

ents. Moreover, they primarily focus on actions with "local" effects, while we permit actions with more general effects.

Son and Baral (2001) describe the notion of an *a-state* or "approximate state" that represents an agent's knowledge by sets of fluents that an agent knows to be true and knows to be false. They also define the notion of a *0-approximation* that provides a simple method for progressing an a-state based on fluents that *must* become known and those that *may* be true or false after an action is applied. Son and Baral provide a translation to situation calculus theories, however, the language they consider is only propositional.

The focus of our work is somewhat different, with an emphasis on formally characterizing the conditions that underlie certain knowledge-restricted theories. Our set of decomposition properties also helps us better understand what we give up representationally in these accounts. Furthermore, this work offers the possibility of identifying action theories that can be compiled into some of the alternative accounts of knowledge we mention above—and extending the compilation approaches to new classes of action theories that haven't been previously considered. For instance, our model of sensing is more general than that of (Petrick and Levesque 2002) and our class of fluent dependent theories is not a class they consider. We show elsewhere that their compilation technique can be extended to these action theories (Petrick 2006). More work is also needed to identify new action theories that preserve the Cartesian property, as the theories we consider aren't exhaustive. Finally, we believe there are practical applications of this work to planning, and we are investigating the relationship between the "standard" representations of domains used by planners that plan with incomplete information and sensing (e.g., Contingent-FF (Hoffmann and Brafman 2005)) and those of knowledge-restricted planners like PKS (Petrick and Bacchus 2002).

Acknowledgements

Special thanks go to Hector Levesque and Yves Lespérance for all of their comments and suggestions during the development of this work. Thanks also to the KR reviewers for their helpful feedback. This work was partially funded by the Natural Sciences and Engineering Research Council (NSERC) scholarship program while the author was at the University of Toronto, and by the European Commission through the PACO-PLUS project (FP6-2004-IST-4-27657).

Appendix (Proof Notes)

Theorem 1 KBATs include a subset of the following initial accessibility axioms:

1. Reflexivity: $(\forall s).\mathit{Init}(s) \supset K(s,s)$,
2. Transitivity: $(\forall s_1,s_2,s_3).\mathit{Init}(s_1) \wedge \mathit{Init}(s_2) \wedge \mathit{Init}(s_3) \supset (K(s_2,s_1) \wedge K(s_3,s_2) \supset K(s_3,s_1))$,
3. Euclidean: $(\forall s_1,s_2,s_3).\mathit{Init}(s_1) \wedge \mathit{Init}(s_2) \wedge \mathit{Init}(s_3) \supset (K(s_2,s_1) \wedge K(s_3,s_1) \supset K(s_2,s_3))$,
4. Symmetry: $(\forall s_1,s_2).\mathit{Init}(s_1) \wedge \mathit{Init}(s_2) \supset (K(s_1,s_2) \supset K(s_2,s_1))$.

The proof seeks to extend these properties from initial situations to all situations, by using the form of the K axiom to argue that a set of K-accessible situations only changes in a non-increasing manner. Thus, when $K(do(a,s'),do(a,s))$ holds it must be the case that $K(s',s)$ also holds. The proof considers the conditions that could hold on successor and predecessor situations, as encoded in K, to show each accessibility relationship is preserved when an action is applied.

Theorem 2 The proof details are mechanical and follow from the definition of **Knows** and ordinary first-order logic.

Theorem 3 Theorem 1 ensures that each set of accessibility properties holds for all situations. We then show that given a Cartesian situation s, s and all situations K-related to s share the same set of accessible situations. Thus, K acts as an equivalence relation over this set. (We can also verify that no single accessibility property is alone sufficient to establish a similar result. See (Petrick 2006) for details.)

Theorem 4 The proof is by induction on n. In the induction step we consider the case where k of the θ_is hold at a single K-related situation, and θ_{k+1} holds at some K-related situation. We then consider the form of θ_{k+1} and argue that for each of the four possible types of pfls, Definition 5 ensures the existence of a K-related situation at which all $k+1$ of the θ_is hold.

Theorem 5 In (1), the $\Leftarrow$ direction follows as a consequence of the standard definition of **Knows**. The $\Rightarrow$ direction uses Theorem 4 and an intermediate result:

$$\Sigma \models (\forall s).\big[\bigwedge\nolimits_{i=1}^{n}(\exists s')(K(s',s) \wedge \phi_i[s']) \supset (\exists s^*)(K(s^*,s) \wedge \bigwedge\nolimits_{i=1}^{n}\phi_i[s^*])\big] \equiv \big[\mathbf{Knows}(\bigvee\nolimits_{i=1}^{n}\neg\phi_i(\mathit{now}),s) \supset \bigvee\nolimits_{i=1}^{n}\mathbf{Knows}(\neg\phi_i(\mathit{now}),s)\big]$$

that holds for arbitrary first-order sentences ϕ_i, observing that the negation of a pfl is also a pfl and that a *DNCI* formula does not change for a collection of pfls if the underlying signs of the pfls change. In (2), the proof is a straightforward application of (1) and the standard properties of **Knows**. In (3), the proof uses (2) and is otherwise a straightforward application of the standard properties of **Knows**. In (4), the $\Leftarrow$ direction follows as a consequence of the ordinary definition and properties of **Knows**. The $\Rightarrow$ direction is established from first principles by arguing from Definition 5. In (5), the result follows from (1) and by arguing that particular *DNCI* formulae do not change if we change the underlying functional pfls from equality to inequality mappings.

Theorem 6 We note that $\Sigma \models (\forall s).\phi[s] \equiv \bigwedge_{i=1}^{n}\pi_i[s]$ follows by expressing the prime implicates π_i of ϕ in Blake Canonical Form. The result then follows from the standard properties of **Knows** and Theorem 5(1).

Theorem 7 We establish this result by reasoning about Definition 5 and the form of the K axiom. Intuitively, we consider two situations s_1 and s_2 that are K-related to a situation $do(\alpha,s)$, where s is Cartesian and α is a sensing action. For relational fluents, if an instance of a fluent holds at s_1 and the negation of the same instance holds at s_2, then we use the property that sensing actions leave fluent values unchanged to conclude that the same instances hold at the predecessor situations of s_1 and s_2 respectively, and

these predecessors are necessarily K-related to s. Using the Cartesian definition we establish the existence of a "cross-product" situation that is K-related to s and then argue from the form of the K axiom (and that it only senses particular instances of fluent literals) that its successor situation must be K-related to $do(\alpha, s)$. Functional fluents are similar.

Theorem 8 The proof is by induction over situations. The approach is similar to the proof of Theorem 7 and we need only consider the case of physical actions (Theorem 7 supplies the result for sensing actions). Given a pair of differing fluent instances at situations K-related to $do(\alpha, s)$, we can reason that α could not have changed these instances, since such changes would apply across the set of K-related situations due to the situation-independent components of the ordinary successor state axioms. We can then apply the Cartesian definition at s and again argue from the form of the ordinary successor state axioms that an appropriate cross-product successor situation must exist.

Theorem 9 The proof is by induction over situations. In the induction step we need only consider the case when a physical action is applied. The tricky part is establishing the existence of the required "cross-product" situation in the case when the fluent in question is part of a dependency chain. We must reason about the existence of a predecessor situation (using the induction assumption) that when progressed through the dependency update, results in the desired configuration of fluents.

Theorem 10 Given the restrictions on sets of propositional pfls without repeated fluent symbols, each *DNCI* formula will be $\top$. The $\Rightarrow$ direction can then be established by Theorem 5(1). In the $\Leftarrow$ direction we make use of the intermediate result from the proof of Theorem 5(1) and a second intermediate result that establishes the converse of Theorem 4. This second result follows from Definition 5 when subjected to the restrictions of proposition KBATs (i.e., no functional fluents and no non-situation arguments).

Theorem 11 The proof of this result follows from Theorem 10 and the intermediate result that

$$\Sigma \models (\forall s).\mathbf{Knows}(\phi(do(\vec{A}, now)), s) \equiv \mathbf{Knows}(\phi(now), do(\vec{A}, s)),$$

where ϕ is a first-order sentence and $\vec{A}$ is a sequence of physical actions. (In temporal logic this is similar to $K \circ^n \phi \equiv \circ^n K\phi$.) The intermediate result itself holds by induction on the length of $\vec{A}$, and by reasoning about the definition of **Knows** and the observation that the K axiom does not change the size of a set of K-related situations when a physical action is applied. (If $K(s', s)$ holds then $K(do(\alpha, s'), do(\alpha, s))$ holds when α is a physical action.)

References

Demolombe, R., and Pozos Parra, M. P. 2000. A simple and tractable extension of situation calculus to epistemic logic. In *Proc. of ISMIS-2000*, 515–524.

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1995. *Reasoning About Knowledge*. MIT Press.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Funge, J. 1998. Interval-valued epistemic fluents. In *AAAI Fall Symposium on Cognitive Robotics*, 23–25.

Hintikka, J. 1962. *Knowledge and Belief*. Ithaca, NY: Cornell University Press.

Hoffmann, J., and Brafman, R. 2005. Contingent planning via heuristic forward search with implicit belief states. In *Proc. of ICAPS-05*, 71–80.

Lin, F., and Reiter, R. 1997. How to progress a database. *Artificial Intelligence* 92(1–2):131–167.

Liu, Y., and Levesque, H. J. 2005. Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions. In *Proc. of IJCAI-05*, 522–527.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence* 4:463–502.

McCarthy, J. 1979a. Ascribing mental qualities to machines. In *Philosophical Perspectives in Artificial Intelligence*. Harvester Press. Reprinted in (1990) *Formalizing Common Sense*, Ablex, Norwood, NJ.

McCarthy, J. 1979b. First order theories of individual concepts and propositions. *Machine Intelligence* 9:129–148.

Moore, R. C. 1985. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*. Norwood, NJ: Ablex Publishing. 319–358.

Petrick, R. P. A., and Bacchus, F. 2002. A knowledge-based approach to planning with incomplete information and sensing. In *Proc. of AIPS-2002*, 212–221. AAAI Press.

Petrick, R. P. A., and Levesque, H. J. 2002. Knowledge equivalence in combined action theories. In *Proc. of KR-2002*, 303–314. Morgan Kaufmann Publishers.

Petrick, R. P. A. 2006. *A knowledge-level approach for effective acting, sensing, and planning*. Ph.D. Thesis, Department of Computer Science, University of Toronto.

Reiter, R. 2001. *Knowledge In Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press.

Scherl, R. B., and Levesque, H. J. 1994. The frame problem and knowledge-producing actions. Technical report, University of Toronto.

Scherl, R. B., and Levesque, H. J. 2003. Knowledge, action, and the frame problem. *Artificial Intelligence* 144(1–2):1–39.

Son, T. C., and Baral, C. 2001. Formalizing sensing actions – a transition function based approach. *Artificial Intelligence* 125(1–2):19–91.

Soutchanski, M. 2001. A correspondence between two different solutions to the projection task with sensing. *Commonsense 2001*.

Vassos, S., and Levesque, H. 2007. Progression of situation calculus action theories with incomplete information. In *Proc. of IJCAI-07*, 2029–2034.