Cartesian Situations and Knowledge Decomposition in the Situation Calculus

Ron Petrick

School of Informatics, University of Edinburgh &
Department of Computer Science, University of Toronto

*With special thanks to Hector Levesque and Yves Lespérance*

Knowledge Representation and Reasoning (KR 2008), Sydney, Australia

19 September 2008
Motivation

• Reasoning about knowledge, sensing, and action in the situation calculus

• Standard approach in the situation calculus: reason about an accessibility relation over a set of possible worlds (Scherl & Levesque, 1994; 2003)
  – Representationally rich
  – Computationally problematic: $n$ atomic formulae $\leadsto 2^n$ possible worlds

• Alternate representations trade expressiveness for tractable reasoning, e.g., model knowledge by sets of fluents known to be true and known to be false

⇒ How do the alternate representations relate to the standard account? Can we restrict the standard account to bring about similar representations? What must we give up to do so?

⇒ Applications to planning?
Overview

- Formal definition of a Cartesian situation in the situation calculus, a structural restriction on standard Scherl & Levesque action theories
- A set of “decomposition” properties that arise from Cartesian situations, for simplifying epistemic formulae into formulae that only mention fluent literals
- Identification of expressive classes of action theories that preserve the Cartesian property through action

⇒ By reducing general expressions of knowledge to knowledge of fluent literals, we can identify standard Scherl & Levesque-style theories that can be compiled into alternative representations that do not require possible worlds. (We do not perform the transformation in this work.)

⇒ Many types of planning actions fall into the classes we consider
Situation calculus and basic action theories

• A logical formalism for modelling dynamic domains
  (McCarthy & Hayes, 1969; Reiter, 2001)

• Actions, situations, fluents (relational, functional), $S_0$, $do(a,s)$

• Basic action theory (BAT)
  1. For each action $A$, an action precondition axiom:
     \[
     \text{Poss}(A(\vec{x}), s) \equiv \top.
     \]
  2. For each relational fluent $F$, a successor state axiom of the form
     \[
     F(\vec{x}, do(a,s)) \equiv \gamma_F^+(\vec{x}, a, s) \lor F(\vec{x}, s) \land \neg \gamma_F^-(\vec{x}, a, s).
     \]
  3. For each functional fluent $f$, a successor state axiom of the form
     \[
     f(\vec{x}, do(a,s)) = y \equiv \gamma_f(\vec{x}, y, a, s) \lor f(\vec{x}, s) = y \land \neg (\exists y') \gamma_f(\vec{x}, y', a, s).
     \]
  4. A set of initial situation axioms
  5. Unique names axioms for actions and a set of foundational axioms.
A $K$ fluent in the situation calculus

- The original situation calculus formalism does not distinguish between what is true in a situation and what is known in a situation.
- Introduce a binary relation $K(s', s)$, meaning “$s'$ is accessible from $s$”

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

- Initial situation must include possible world alternatives

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

- A (modified) successor state axiom for $K$ (see paper for details)

$\Rightarrow$ KBAT
Example: $K$ successor state axiom

\[
K(s'', do(a, s)) \equiv (\exists s'). s'' = do(a, s') \land K(s', s) \land \left( \forall x \right) (a = \text{sense}_1(x) \supset (F(x, s) \equiv F(x, s'))) \land \left( \forall x \right) (a = \text{sense}_2 \supset (F(x, s) \equiv F(x, s'))) \land \left( \forall x \right) (a = \text{sense}_3(x) \supset (f(x, s) = f(x, s')) \land (f(x, s) = c \supset (F(x, s) \equiv F(x, s')))).
\]

- $\text{sense}_1(x)$ unconditionally senses $F(x)$
- $\text{sense}_2$ unconditionally senses $F(x)$ for all $x$
- $\text{sense}_3(x)$ unconditionally senses $f(x)$ and conditionally senses $F(x)$, provided $f(x, s) = c$
Basic properties of Knows

Theorem

Let $\Sigma$ be a KBAT. If $\phi$ and $\psi$ are first-order sentences and $\phi(\vec{x},s)$ is a first-order formula then

1. $\Sigma \models (\forall s). (\text{Knows}(\phi(\text{now}),s) \land \text{Knows}(\psi(\text{now}),s)) \equiv \text{Knows}(\phi(\text{now}) \land \psi(\text{now}),s)$,

2. $\Sigma \models (\forall s). (\text{Knows}(\phi(\text{now}),s) \lor \text{Knows}(\psi(\text{now}),s)) \supset \text{Knows}(\phi(\text{now}) \lor \psi(\text{now}),s)$,

3. $\Sigma \models (\forall s). (\forall \vec{x}) \text{Knows}(\phi(\vec{x},\text{now}),s) \equiv \text{Knows}((\forall \vec{x}) \phi(\vec{x},\text{now}),s)$.

4. $\Sigma \models (\forall s). (\exists \vec{x}) \text{Knows}(\phi(\vec{x},\text{now}),s) \supset \text{Knows}((\exists \vec{x}) \phi(\vec{x},\text{now}),s)$.

• KBATs + restrictions $\rightsquigarrow$ knowledge-level properties?
Cartesian products and situations

• A Cartesian product structure on a set of possible worlds introduces an independence property that removes dependencies between individual fluents (McCarthy, 1979)

• Cartesian product over sets

{a, b} × {c, d} = {〈a, c〉, 〈b, c〉, 〈b, d〉, 〈a, d〉}

• Cartesian product over situations

![Cartesian product over situations](image)
Cartesian situations defined

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $\Sigma$ be a KBAT with a finite number of fluent symbols. Let $F_1, F_2, \ldots, F_n$ be the relational fluent symbols of $\Sigma$, and let $f_1, f_2, \ldots, f_m$ be the functional fluent symbols of $\Sigma$. The abbreviation $Cart(s)$ is defined as the expression:</td>
</tr>
</tbody>
</table>

\[
Cart(s) \overset{\text{def}}{=} \left[ \bigwedge_{i=1}^{n} (\forall s', s'', \vec{x}_i).K(s', s) \land K(s'', s) \land F_i(\vec{x}_i, s') \neq F_i(\vec{x}_i, s'') \supset \right. \\
\left. (\exists s^*).K(s^*, s) \land F_i(\vec{x}_i, s^*) \equiv F_i(\vec{x}_i, s'') \land (\forall \vec{x}_j) (\vec{x}_j \neq \vec{x}_i \supset F_i(\vec{x}_j, s^*) \equiv F_i(\vec{x}_j, s')) \land \right. \\
\left. \bigwedge_{k=1}^{n} (\forall \vec{x}_k) F_k(\vec{x}_k, s^*) \equiv F_k(\vec{x}_k, s') \land \right. \\
\left. \bigwedge_{k=1}^{m} (\forall \vec{x}_k) f_k(\vec{x}_k, s^*) = f_k(\vec{x}_k, s') \right] \land \\
\ldots
\]
Cartesian situations

- Cartesian situations place strong structural restrictions on $K$-related situations
- Certain configurations of situations are prohibited
  
  E.g., consider representations of $\text{Knows}(F(\text{now}) \lor \neg G(\text{now}), s)$

- What does this mean in terms of “knowledge level” properties?
- Focus on collections of primitive fluent literals (pfls)

\[
F(\bar{x}, s), \quad f(\bar{x}) = y, \quad \neg F(\bar{x}, s), \quad f(\bar{x}) \neq t
\]
Properties of Cartesian situations

**Theorem**

Let $\Sigma$ be a KBAT and let the $\theta_i$s be a collection of pfls.

1. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n). Cart(s) \land DNCI(\theta_1, \theta_2, \ldots, \theta_n) \supset$

   $\left[ \text{Knows}(\bigvee_{i=1}^n \theta_i(\text{now}), s) \equiv \bigvee_{i=1}^n \text{Knows}(\theta_i(\text{now}), s) \right]$

2. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n). Cart(s) \land DNCI(\theta_1, \theta_2, \ldots, \theta_n) \supset$

   $\left[ \text{Knows}(\bigwedge_{i=1}^{n-1} \theta_i(\text{now}) \equiv \theta_{i+1}(\text{now}), s) \equiv$

   $\bigwedge_{i=1}^n \text{Knows}(\theta_i(\text{now}), s) \lor \bigwedge_{i=1}^n \text{Knows}(\overline{\theta}_i(\text{now}), s) \right]$
Properties of Cartesian situations...(2)

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
</table>
| 3. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n).\text{Cart}(s) \land \text{DNCl}(f_1(\vec{x}_1), f_2(\vec{x}_2), \ldots, f_n(\vec{x}_n)) \supset$
| $\left[ \text{Knows}(\bigwedge_{i=1}^{n-1} f_i(\vec{x}_i, \text{now}) = f_{i+1}(\vec{x}_{i+1}, \text{now}), s) \equivight.$
| $(\exists y). \bigwedge_{i=1}^{n} \text{Knows}(f_i(\vec{x}_i, \text{now}) = y, s)]$ |

| 4. $\Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n).\text{Cart}(s) \land \text{DNCl}(f_1(\vec{x}_1), f_2(\vec{x}_2), \ldots, f_n(\vec{x}_n)) \supset$
| $\left[ \text{Knows}(\bigvee_{i=1}^{n-1} f_i(\vec{x}_i, \text{now}) \neq f_{i+1}(\vec{x}_{i+1}, \text{now}), s) \equivight.$
| $(\forall y) \bigvee_{i=1}^{n} \text{Knows}(f_i(\vec{x}_i, \text{now}) \neq y, s)].$ |
Properties of Cartesian situations...(3)

Note that property

\[ 3. \quad \Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n). Cart(s) \land \text{DNCl}(f_1(\vec{x}_1), f_2(\vec{x}_2), \ldots, f_n(\vec{x}_n)) \supset\]

\[ \left[ \text{Knows}(\bigwedge_{i=1}^{n-1} f_i(\vec{x}_i, \text{now}) = f_{i+1}(\vec{x}_{i+1}, \text{now}), s) \right. \]

\[ \left. \equiv (\exists y). \bigwedge_{i=1}^{n} \text{Knows}(f_i(\vec{x}_i, \text{now}) = y, s) \right] \]

is equivalent to

\[ 3'. \quad \Sigma \models (\forall s, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n). Cart(s) \land \text{DNCl}(f_1(\vec{x}_1), f_2(\vec{x}_2), \ldots, f_n(\vec{x}_n)) \supset\]

\[ \left[ \text{Knows}(\exists y. \bigwedge_{i=1}^{n} f_i(\vec{x}_i, \text{now}) = y, s) \right. \]

\[ \left. \equiv (\exists y). \bigwedge_{i=1}^{n} \text{Knows}(f_i(\vec{x}_i, \text{now}) = y, s) \right] . \]
Example: applying Cartesian properties

• Using Cartesian properties we can decompose certain types of epistemic formulae into “simpler” epistemic formulae that only mention fluent literals

\[
\Sigma \models \text{Knows}(\phi(\text{now}), s)
\]

\[
\Rightarrow \Sigma \models \text{Knows}(\bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m} \alpha_{ij}(\text{now})), s)
\]

\[
\Rightarrow \Sigma \models \bigwedge_{i=1}^{n} \text{Knows}(\bigvee_{j=1}^{m} \alpha_{ij}(\text{now}), s)
\]

\[
\Rightarrow \Sigma \models \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} \text{Knows}(\alpha_{ij}(\text{now}), s).
\]

• Cartesian properties come at a representational cost
  – Certain types of knowledge can’t be represented
  – E.g., require definite knowledge of particular components of a disjunction
Progressing Cartesian situations through action

- Can we identify expressive KBATs that preserve the Cartesian property through action? Yes
- Knowledge-producing actions preserve the Cartesian property

Theorem

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

(This theorem does not necessarily hold if we change the form of the $K$ axiom)

- Physical actions are more problematic since the Cartesian property is “brittle”
- Easy to construct physical actions that break the Cartesian property

$\Rightarrow$ Form classes of KBATs by restricting the ordinary successor state axioms
Context-free theories

• Each successor state axiom has the form

\[ F(\vec{x}, do(a, s)) \equiv \gamma_F^+(\vec{x}, a) \lor F(\vec{x}, s) \land \neg \gamma_F^-(\vec{x}, a), \]

for a relational fluent \( F \), or

\[ f(\vec{x}, do(a, s)) = y \equiv \gamma_f(\vec{x}, y, a) \lor f(\vec{x}, s) = y \land \neg (\exists y') \gamma_f(\vec{x}, y', a), \]

for a functional fluent \( f \), where \( \gamma_F^+, \gamma_F^-, \) and \( \gamma_f \) are all situation independent.

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>If ( \Sigma ) is context free then ( \Sigma \models (\forall s)(\text{Init}(s) \supset \text{Cart}(s)) \supset (\forall s)\text{Cart}(s). )</em></td>
</tr>
</tbody>
</table>

• STRIPS-style actions (Fikes & Nilsson, 1971), i.e., actions with unconditional effects, are a subset of this class.
Fluent-dependent theories

• Allow limited fluent references in the effect portions of successor state axioms
  – “Cyclic” dependency
    \[
    F(do(a, s)) \equiv (a = A \land G(s)) \lor F(s) \land \neg(a = A \land \neg G(s))
    \]
    \[
    G(do(a, s)) \equiv (a = A \land F(s)) \lor G(s) \land \neg(a = A \land \neg F(s)).
    \]
  – “Linear” dependency
    \[
    F(do(a, s)) \equiv (a = A \land G(s)) \lor F(s) \land \neg(a = A \land \neg G(s))
    \]
    \[
    G(do(a, s)) \equiv a = A \lor G(s).
    \]
    (See the paper for the general syntactic form)

• Action produce “symmetric” behaviour across the set of K-related situations

Theorem

If \(\Sigma\) is fluent dependent then \(\Sigma \models (\forall s)(Init(s) \supset Cart(s)) \supset (\forall s)Cart(s)\).
Propositional theories (pKBATs)

- Ordinary relational fluents only have situation arguments; no functional fluents
- Requirements for Cartesian situations become much simpler

Theorem

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

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

- Provided no action sequence gives rise to particular epistemic disjunctions that can’t be decomposed, the resulting situation will be Cartesian
- Theories with finite domains of discourse can be converted to this form (cf. many planning algorithm first “propositionalize” domains)
Example: Unix domain

\[ indir(f, d, do(a, s)) \equiv indir(f, d, s), \]
\[ size(f, do(a, s)) = y \equiv size(f, s) = y, \]
\[ readable(f, do(a, s)) \equiv a = chmod+r(f) \lor \]
\[ readable(f, s) \land \neg(a = chmod-r(f)), \]
\[ K(s'', do(a, s)) \equiv (\exists s').s'' = do(a, s') \land K(s', s) \land \]
\[ (\forall d).(a = ls(d) \lor \]
\[ (\forall f) \ (indir(f, d, s) \equiv indir(f, d, s')) \land \]
\[ (\forall f) \ (indir(f, d, s) \lor (size(f, s) = size(f, s')))) \land \]
\[ (\forall f) \ (indir(f, d, s) \lor (readable(f, s) \equiv readable(f, s')))). \]

- \textit{chmod+r} (\textit{chmod-r}) unconditionally makes file \( f \) readable (unreadable)
- \textit{ls} conditionally senses the size and readability of all files in directory \( d \)
Example: a FIFO buffer

\[ B_1(\text{do}(a,s)) = y \equiv (\exists d)(a = \text{write}(d) \land y = d) \lor \]
\[ B_1(s) = y \land \neg(\exists y')(\exists d).a = \text{write}(d) \land y' = d), \]
\[ B_2(\text{do}(a,s)) = y \equiv ((\exists d).a = \text{write}(d) \land B_1(s) = y) \lor \]
\[ B_2(s) = y \land \neg(\exists y')(\exists d).a = \text{write}(d) \land B_1(s) = y'), \]
\[ B_3(\text{do}(a,s)) = y \equiv ((\exists d).a = \text{write}(d) \land B_2(s) = y) \lor \]
\[ B_3(s) = y \land \neg(\exists y')(\exists d).a = \text{write}(d) \land B_2(s) = y'). \]
Conclusions

• Cartesian situations give rise to a set of properties that let us decompose epistemic formulae into more primitive expressions of fluent literal knowledge (additional properties in paper, e.g., Cartesian situations and $K$ accessibility)

• These properties come at a representational cost, most notably disjunctions

• We have identified expressive classes of KBATs that preserve the Cartesian property through action (future work: identify new classes)

• We can use the Cartesian property to find theories that can be compiled into alternative representations with similar restrictions but tractable reasoning

  – Knowledge fluents: $F, G, K \rightsquigarrow KF, K\neg F, KG, K\neg G$ (Demolombe & Pozos Parra, 2000; Petrick & Levesque, 2002; Petrick, 2006)


• Current focus: syntactic transformation of actions into a knowledge-level form usable by the PKS planner (Petrick & Bacchus, 2002)


