Discrete Event Calculus with Branching Time

Erik T. Mueller
IBM Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598 USA

Abstract

We add branching time to the linear discrete event calculus, which yields a formalism for commonsense reasoning that combines the benefits of the situation calculus and the event calculus. We show how the branching discrete event calculus can be used to solve commonsense reasoning problems involving hypothetical events, concurrent events with cumulative and canceling effects, and triggered events.

Introduction

The classical logic event calculus (Miller & Shanahan 2002; Shanahan 1997) can serve as a foundation for commonsense reasoning. It can be used to reason about important areas of the commonsense world including action and change (Shanahan 1999a), space (Morgenstern 2001; Shanahan 1996; 2004), and mental states (Mueller 2006a). An important aspect of commonsense reasoning is reasoning about hypothetical events. Unlike the situation calculus (McCarthy & Hayes 1969), the classical logic event calculus typically uses a linear time structure (Miller & Shanahan 2002, p. 453) and does not handle hypothetical events (Shanahan 1997, p. 364).

In this paper, we show how a version of the classical logic event calculus can be modified to yield a new formalism that combines the benefits of the situation calculus and the event calculus. Like the situation calculus, the new formalism supports reasoning about hypothetical events. Like the classical logic event calculus, the new formalism supports reasoning about the commonsense law of inertia, release from the commonsense law of inertia, concurrent events with cumulative and canceling effects, context-sensitive effects, indirect effects, nondeterministic effects, preconditions, and triggered events.

We start with the linear discrete event calculus (LDEC) (Mueller 2004a; 2006a), a discrete version of the classical logic event calculus. LDEC has been proved logically equivalent to the continuous event calculus for integer time (Mueller 2004a), and to temporal action logics (Doherty *et al.* 1998) for inertial fluents and single-step actions (Mueller 2006b). We modify LDEC to obtain the branching discrete event calculus (BDEC) by (1) removing the requirement that every situation must have a unique successor, and (2) adding an argument for successor situation to *Happens*, *Initiates*, *Terminates*, and *Releases*. We show how BDEC can be used to solve commonsense reasoning problems involving hypothetical events, concurrent events, and triggered events. We extend BDEC to distinguish between hypothetical and actual situations and events. In an extended version of this paper (Mueller in submission), we characterize the precise relationship between LDEC and BDEC, and prove that a restricted version of BDEC is equivalent to a version of the situation calculus.

The discrete event calculus was developed to facilitate automated event calculus reasoning. It simplifies the classical logic event calculus axioms, reducing the number of axioms from 17 to 12 and eliminating triply quantified time from most axioms. The discrete event calculus is the basis for the Discrete Event Calculus Reasoner program for automated commonsense reasoning (Mueller 2004b).[1] We have implemented BDEC within this program, extending it with the ability to reason about hypothetical events.

Linear Discrete Event Calculus

We use many-sorted languages with equality. The linear discrete event calculus has sorts for events, fluents, and situations. We use a version of the linear discrete event calculus without gradual change, and with an axiomatization of the nonnegative integers. The language has the constant S_0 denoting the initial situation, the function $S_L(s)$, which denotes the unique successor of situation s, and the following predicates:

- *Happens*$_L(e, s)$: Event e occurs at situation s.
- *HoldsAt*(f, s): Fluent f is true at situation s.
- *ReleasedAt*(f, s): Fluent f is released from the commonsense law of inertia at situation s.
- *Initiates*$_L(e, f, s)$: If event e occurs at situation s, then fluent f will be true and not released from the commonsense law of inertia at the successor of s.
- *Terminates*$_L(e, f, s)$: If event e occurs at situation s, then fluent f will be false and not released from the commonsense law of inertia at the successor of s.
- *Releases*$_L(e, f, s)$: If event e occurs at situation s, then fluent f will be released from the commonsense law of inertia at the successor of s.

The commonsense law of inertia (Shanahan 1997) states that a fluent's truth value persists unless the fluent is affected by

[1]This program is available for download at `http://decreasoner.sourceforge.net/`.

an event. When a fluent is released from this law, its truth value can fluctuate. Fluents that are released from the commonsense law of inertia can be used to model nondeterministic effects (Shanahan 1999a) and indirect effects (Shanahan 1999b).

Let LDEC be the conjunction of the following axioms:
LDEC1. $S_L(s) \neq S_0$
LDEC2. $S_L(s_1) = S_L(s_2) \rightarrow s_1 = s_2$
LDEC3. $\forall P\,((P(S_0) \wedge \forall s\,(P(s) \rightarrow P(S_L(s)))) \rightarrow \forall s\, P(s))$
LDEC4. $\mathit{HoldsAt}(f,s) \wedge \neg \mathit{ReleasedAt}(f, S_L(s)) \wedge$
$\neg\exists e\,(\mathit{Happens}_L(e,s) \wedge \mathit{Terminates}_L(e,f,s)) \rightarrow$
$\mathit{HoldsAt}(f, S_L(s))$
LDEC5. $\neg\mathit{HoldsAt}(f,s) \wedge \neg\mathit{ReleasedAt}(f,S_L(s)) \wedge$
$\neg\exists e\,(\mathit{Happens}_L(e,s) \wedge \mathit{Initiates}_L(e,f,s)) \rightarrow$
$\neg\mathit{HoldsAt}(f,S_L(s))$
LDEC6. $\mathit{ReleasedAt}(f,s) \wedge$
$\neg\exists e\,(\mathit{Happens}_L(e,s) \wedge (\mathit{Initiates}_L(e,f,s) \vee$
$\mathit{Terminates}_L(e,f,s))) \rightarrow \mathit{ReleasedAt}(f,S_L(s))$
LDEC7. $\neg\mathit{ReleasedAt}(f,s) \wedge$
$\neg\exists e\,(\mathit{Happens}_L(e,s) \wedge \mathit{Releases}_L(e,f,s)) \rightarrow$
$\neg\mathit{ReleasedAt}(f,S_L(s))$
LDEC8. $\mathit{Happens}_L(e,s) \wedge \mathit{Initiates}_L(e,f,s) \rightarrow$
$\mathit{HoldsAt}(f,S_L(s))$
LDEC9. $\mathit{Happens}_L(e,s) \wedge \mathit{Terminates}_L(e,f,s) \rightarrow$
$\neg\mathit{HoldsAt}(f,S_L(s))$
LDEC10. $\mathit{Happens}_L(e,s) \wedge \mathit{Releases}_L(e,f,s) \rightarrow$
$\mathit{ReleasedAt}(f,S_L(s))$
LDEC11. $\mathit{Happens}_L(e,s) \wedge$
$(\mathit{Initiates}_L(e,f,s) \vee \mathit{Terminates}_L(e,f,s)) \rightarrow$
$\neg\mathit{ReleasedAt}(f,S_L(s))$

Axioms LDEC1 through LDEC3 are the Peano axioms for the nonnegative integers. Axiom LDEC3 is a second-order axiom of induction.

Branching Discrete Event Calculus

We extend LDEC to branching time as follows. We replace the function S_L with a relation *S*, and allow a situation to have zero or more successors. We add an argument to *Happens*, *Initiates*, *Terminates*, and *Releases* specifying the successor situation. It is important to add the successor situation to *Initiates*, *Terminates*, and *Releases* to treat concurrent events with cumulative and canceling effects (see below).

The language has the constant S_0 denoting the initial situation, and the following predicates:

• $S(s_1, s_2)$: Situation s_2 is a successor of situation s_1.
• $\mathit{Happens}(e, s_1, s_2)$: Event e occurs between situation s_1 and situation s_2.
• $\mathit{HoldsAt}(f, s)$: Fluent f is true at situation s.
• $\mathit{ReleasedAt}(f, s)$: Fluent f is released from the commonsense law of inertia at situation s.
• $\mathit{Initiates}(e, f, s_1, s_2)$: If event e occurs between situation s_1 and situation s_2, then fluent f will be true and not released from the commonsense law of inertia at s_2.
• $\mathit{Terminates}(e, f, s_1, s_2)$: If event e occurs between situation s_1 and situation s_2, then fluent f will be false and not released from the commonsense law of inertia at s_2.
• $\mathit{Releases}(e, f, s_1, s_2)$: If event e occurs between situation s_1 and situation s_2, then fluent f will be released from the commonsense law of inertia at s_2.

Let BDEC be the conjunction of the following axioms:
BDEC1. $\neg S(s, S_0)$
BDEC2. $S(s_1, s) \wedge S(s_2, s) \rightarrow s_1 = s_2$
BDEC3. $\forall P\,((P(S_0) \wedge \forall s_1, s_2\,(S(s_1,s_2) \wedge P(s_1) \rightarrow P(s_2))) \rightarrow$
$\forall s\, P(s))$
BDEC4. $S(s_1,s_2) \wedge \mathit{HoldsAt}(f,s_1) \wedge \neg\mathit{ReleasedAt}(f,s_2) \wedge$
$\neg\exists e\,(\mathit{Happens}(e,s_1,s_2) \wedge \mathit{Terminates}(e,f,s_1,s_2)) \rightarrow$
$\mathit{HoldsAt}(f,s_2)$
BDEC5. $S(s_1,s_2) \wedge \neg\mathit{HoldsAt}(f,s_1) \wedge \neg\mathit{ReleasedAt}(f,s_2) \wedge$
$\neg\exists e\,(\mathit{Happens}(e,s_1,s_2) \wedge \mathit{Initiates}(e,f,s_1,s_2)) \rightarrow$
$\neg\mathit{HoldsAt}(f,s_2)$
BDEC6. $S(s_1,s_2) \wedge \mathit{ReleasedAt}(f,s_1) \wedge$
$\neg\exists e\,(\mathit{Happens}(e,s_1,s_2) \wedge$
$(\mathit{Initiates}(e,f,s_1,s_2) \vee \mathit{Terminates}(e,f,s_1,s_2))) \rightarrow$
$\mathit{ReleasedAt}(f,s_2)$
BDEC7. $S(s_1,s_2) \wedge \neg\mathit{ReleasedAt}(f,s_1) \wedge$
$\neg\exists e\,(\mathit{Happens}(e,s_1,s_2) \wedge \mathit{Releases}(e,f,s_1,s_2)) \rightarrow$
$\neg\mathit{ReleasedAt}(f,s_2)$
BDEC8. $\mathit{Happens}(e,s_1,s_2) \wedge \mathit{Initiates}(e,f,s_1,s_2) \rightarrow$
$\mathit{HoldsAt}(f,s_2)$
BDEC9. $\mathit{Happens}(e,s_1,s_2) \wedge \mathit{Terminates}(e,f,s_1,s_2) \rightarrow$
$\neg\mathit{HoldsAt}(f,s_2)$
BDEC10. $\mathit{Happens}(e,s_1,s_2) \wedge \mathit{Releases}(e,f,s_1,s_2) \rightarrow$
$\mathit{ReleasedAt}(f,s_2)$
BDEC11. $\mathit{Happens}(e,s_1,s_2) \wedge$
$(\mathit{Initiates}(e,f,s_1,s_2) \vee \mathit{Terminates}(e,f,s_1,s_2)) \rightarrow$
$\neg\mathit{ReleasedAt}(f,s_2)$
BDEC12. $\mathit{Happens}(e,s_1,s_2) \rightarrow S(s_1,s_2)$

Axioms BDEC1 through BDEC3 are generalized Peano axioms along the lines of Schmidt (1960), which do not require each situation to have exactly one successor. BDEC3 is a second-order induction axiom similar to the one used by Reiter (1993) for the situation calculus. In any model of these axioms, situations form a tree whose root is S_0.[2]

Commonsense Reasoning Problems

BDEC can be used to reason about (1) hypothetical events as in the situation calculus, and (2) phenomena of action and change as in the event calculus. In this section, we show how BDEC can be used to perform commonsense reasoning about three scenarios: the hypothetical Yale shooting scenario, the soup bowl scenario, and the reactive cat scenario.

Hypothetical Yale Shooting Scenario Van Belleghem, Denecker, and De Schreye (1997) describe the following problem of hypothetical reasoning, which is based on the Yale shooting scenario (Hanks & McDermott 1987). A turkey, which was initially alive, was shot by a person. It is not known whether the gun was loaded. But it is known that, if the person had waited instead of shooting, then the gun would have been loaded afterward. The problem is to infer that the gun was initially loaded, and that the turkey died. Van Belleghem et al. argue that this problem can be solved using the situation calculus, but not by the event calculus. We show how BDEC can be used to solve this problem.

We use a domain theory similar to that of Shanahan (1997, pp. 322–323). We have three events, *Load*, *Shoot*, and *Wait*,

[2]See the proof of Proposition 2.1 of Pinto (1994, pp. 103–105).

and two fluents, *Alive* and *Loaded*. If a gun is loaded, then it will be loaded:

$$\mathit{Initiates}(\mathit{Load}, \mathit{Loaded}, s_1, s_2) \tag{1}$$

If a gun is loaded and the gun is shot, then the victim will no longer be alive:

$$\mathit{HoldsAt}(\mathit{Loaded}, s_1) \rightarrow \mathit{Terminates}(\mathit{Shoot}, \mathit{Alive}, s_1, s_2) \tag{2}$$

If a gun is shot, then it will no longer be loaded:

$$\mathit{Terminates}(\mathit{Shoot}, \mathit{Loaded}, s_1, s_2) \tag{3}$$

We consider the following narrative. The victim is initially alive:

$$\mathit{HoldsAt}(\mathit{Alive}, S_0) \tag{4}$$

The gun is shot between situation S_0 and situation S_1:

$$\mathit{Happens}(\mathit{Shoot}, S_0, S_1) \tag{5}$$

We add the following hypothetical information. If the person had waited between situation S_0 and situation S_2, then the gun would have been loaded at S_2:

$$\mathit{Happens}(\mathit{Wait}, S_0, S_2) \tag{6}$$
$$\mathit{HoldsAt}(\mathit{Loaded}, S_2) \tag{7}$$

The events and fluents are distinct:

$$\mathit{Load} \neq \mathit{Shoot} \tag{8}$$
$$\mathit{Shoot} \neq \mathit{Wait} \tag{9}$$
$$\mathit{Load} \neq \mathit{Wait} \tag{10}$$
$$\mathit{Alive} \neq \mathit{Loaded} \tag{11}$$

Fluents are never released from the commonsense law of inertia:

$$\neg \mathit{ReleasedAt}(f, s) \tag{12}$$

We can then show that the gun was loaded at S_0 and that the victim was dead at S_1.

Just as in the classical logic event calculus, we use the nonmonotonic method of circumscription (Lifschitz 1994; McCarthy 1980) for default reasoning about time. We circumscribe *Initiates*, *Terminates*, and *Releases* to minimize unexpected effects of events, and we circumscribe *Happens* to minimize unexpected events.

Proposition 1. Let $\Sigma = (1) \wedge (2) \wedge (3)$, $\Delta = (5) \wedge (6)$, $\Omega = (8) \wedge (9) \wedge (10) \wedge (11)$, and $\Gamma = (4) \wedge (7) \wedge (12)$. Then we have

$$\begin{gathered} \mathit{CIRC}[\Sigma; \mathit{Initiates}, \mathit{Terminates}, \mathit{Releases}] \wedge \\ \mathit{CIRC}[\Delta; \mathit{Happens}] \wedge \Omega \wedge \Gamma \wedge \mathrm{BDEC} \\ \vdash \mathit{HoldsAt}(\mathit{Loaded}, S_0) \wedge \neg \mathit{HoldsAt}(\mathit{Alive}, S_1). \end{gathered}$$

Proof. From $\mathit{CIRC}[\Sigma; \mathit{Initiates}, \mathit{Terminates}, \mathit{Releases}]$ and Propositions 2 and 14 of Lifschitz (1994) reducing circumscription to predicate completion and reducing parallel circumscription to basic circumscription, we have

$$\mathit{Initiates}(e, f, s_1, s_2) \leftrightarrow (e = \mathit{Load} \wedge f = \mathit{Loaded}) \tag{13}$$

$$\begin{gathered} \mathit{Terminates}(e, f, s_1, s_2) \leftrightarrow \\ (e = \mathit{Shoot} \wedge f = \mathit{Alive} \wedge \mathit{HoldsAt}(\mathit{Loaded}, s_1)) \vee \\ (e = \mathit{Shoot} \wedge f = \mathit{Loaded}) \end{gathered} \tag{14}$$

$$\neg \mathit{Releases}(e, f, s_1, s_2) \tag{15}$$

From $\mathit{CIRC}[\Delta; \mathit{Happens}]$ and Proposition 2 of Lifschitz, we have

$$\begin{gathered} \mathit{Happens}(e, s_1, s_2) \leftrightarrow \\ (e = \mathit{Shoot} \wedge s_1 = S_0 \wedge s_2 = S_1) \vee \\ (e = \mathit{Wait} \wedge s_1 = S_0 \wedge s_2 = S_2) \end{gathered} \tag{16}$$

Seeking a contradiction, suppose that

$$\neg \mathit{HoldsAt}(\mathit{Loaded}, S_0) \tag{17}$$

From (13), (16), (8), and (10), we have $\neg \exists e\, (\mathit{Happens}(e, S_0, S_2) \wedge \mathit{Initiates}(e, \mathit{Loaded}, S_0, S_2))$. From this, $S(S_0, S_2)$ (which follows from (16) and BDEC12), (17), (12), and BDEC5, we have $\neg \mathit{HoldsAt}(\mathit{Loaded}, S_2)$, which contradicts (7). Therefore, $\mathit{HoldsAt}(\mathit{Loaded}, S_0)$.

From this and (14), we have $\mathit{Terminates}(\mathit{Shoot}, \mathit{Alive}, S_0, S_1)$. From this, $\mathit{Happens}(\mathit{Shoot}, S_0, S_1)$ (which follows from (16)), and BDEC9, we have $\neg \mathit{HoldsAt}(\mathit{Alive}, S_1)$. ☐

Soup Bowl Scenario Gelfond, Lifschitz, and Rabinov (1991) describe the following soup bowl scenario. A person is trying to lift a bowl of soup. The problem is to infer that, if the person lifts the bowl with one hand, then the soup spills, whereas, if the person lifts the bowl with both hands, then the soup does not spill. Miller and Shanahan (2002, pp. 460–461) have formalized this problem in the classical logic event calculus. We show that their formalization works in BDEC as well. By using BDEC, we are able to consider two hypothetical alternatives.

If the bowl is lifted with both hands, then it will be raised:

$$\mathit{Happens}(\mathit{LiftLeft}, s_1, s_2) \rightarrow \mathit{Initiates}(\mathit{LiftRight}, \mathit{Raised}, s_1, s_2) \tag{18}$$

If the bowl is only lifted with one hand, then it will be spilled:

$$\neg \mathit{Happens}(\mathit{LiftRight}, s_1, s_2) \rightarrow \mathit{Initiates}(\mathit{LiftLeft}, \mathit{Spilled}, s_1, s_2) \tag{19}$$

$$\neg \mathit{Happens}(\mathit{LiftLeft}, s_1, s_2) \rightarrow \mathit{Initiates}(\mathit{LiftRight}, \mathit{Spilled}, s_1, s_2) \tag{20}$$

Initially, the bowl is not raised and not spilled:

$$\neg \mathit{HoldsAt}(\mathit{Raised}, S_0) \tag{21}$$
$$\neg \mathit{HoldsAt}(\mathit{Spilled}, S_0) \tag{22}$$

We consider two alternatives. The first alternative is that the bowl is lifted with both hands:

$$\mathit{Happens}(\mathit{LiftLeft}, S_0, S_1) \tag{23}$$
$$\mathit{Happens}(\mathit{LiftRight}, S_0, S_1) \tag{24}$$

We can show that the bowl will be raised and not spilled. The second alternative is that the bowl is lifted with the right hand:

$$\mathit{Happens}(\mathit{LiftRight}, S_0, S_2) \tag{25}$$

We can show that the bowl will be spilled and not raised.

The events and fluents are distinct:

$$\mathit{LiftLeft} \neq \mathit{LiftRight} \tag{26}$$
$$\mathit{Raised} \neq \mathit{Spilled} \tag{27}$$

Situations S_1 and S_2 are distinct:

$$S_1 \neq S_2 \tag{28}$$

Fluents are never released from the commonsense law of inertia:

$$\neg \mathit{ReleasedAt}(f, s) \tag{29}$$

Proposition 2. Let $\Sigma = (18) \wedge (19) \wedge (20)$, $\Delta = (23) \wedge (24) \wedge (25)$, $\Omega = (26) \wedge (27) \wedge (28)$, and $\Gamma = (21) \wedge (22) \wedge (29)$. Then we have

$$\begin{gathered}\mathit{CIRC}[\Sigma; \mathit{Initiates}, \mathit{Terminates}, \mathit{Releases}] \wedge \\ \mathit{CIRC}[\Delta; \mathit{Happens}] \wedge \Omega \wedge \Gamma \wedge \text{BDEC} \\ \vdash \mathit{HoldsAt}(\mathit{Raised}, S_1) \wedge \neg \mathit{HoldsAt}(\mathit{Spilled}, S_1) \wedge \\ \mathit{HoldsAt}(\mathit{Spilled}, S_2) \wedge \neg \mathit{HoldsAt}(\mathit{Raised}, S_2).\end{gathered}$$

Proof. From $\mathit{CIRC}[\Sigma; \mathit{Initiates}, \mathit{Terminates}, \mathit{Releases}]$ and Propositions 2 and 14 of Lifschitz (1994), we have

$$\begin{gathered}\mathit{Initiates}(e, f, s_1, s_2) \leftrightarrow \\ (e = \mathit{LiftRight} \wedge f = \mathit{Raised} \wedge \\ \mathit{Happens}(\mathit{LiftLeft}, s_1, s_2)) \ \vee \\ (e = \mathit{LiftLeft} \wedge f = \mathit{Spilled} \wedge \\ \neg \mathit{Happens}(\mathit{LiftRight}, s_1, s_2)) \ \vee \\ (e = \mathit{LiftRight} \wedge f = \mathit{Spilled} \wedge \\ \neg \mathit{Happens}(\mathit{LiftLeft}, s_1, s_2))\end{gathered} \tag{30}$$
$$\neg \mathit{Terminates}(e, f, s_1, s_2) \tag{31}$$
$$\neg \mathit{Releases}(e, f, s_1, s_2) \tag{32}$$

From $\mathit{CIRC}[\Delta; \mathit{Happens}]$ and Proposition 2 of Lifschitz, we have

$$\begin{gathered}\mathit{Happens}(e, s_1, s_2) \leftrightarrow \\ (e = \mathit{LiftLeft} \wedge s_1 = S_0 \wedge s_2 = S_1) \ \vee \\ (e = \mathit{LiftRight} \wedge s_1 = S_0 \wedge s_2 = S_1) \ \vee \\ (e = \mathit{LiftRight} \wedge s_1 = S_0 \wedge s_2 = S_2)\end{gathered} \tag{33}$$

From $\mathit{Happens}(\mathit{LiftLeft}, S_0, S_1)$ (which follows from (33)) and (30), we have $\mathit{Initiates}(\mathit{LiftRight}, \mathit{Raised}, S_0, S_1)$. From this, $\mathit{Happens}(\mathit{LiftRight}, S_0, S_1)$ (which follows from (33)), and BDEC8, we have $\mathit{HoldsAt}(\mathit{Raised}, S_1)$.

From (30), (33), (26), (27), and (28), we have $\neg \exists e \, (\mathit{Happens}(e, S_0, S_1) \ \wedge \ \mathit{Initiates}(e, \mathit{Spilled}, S_0, S_1))$. From this, $S(S_0, S_1)$ (which follows from (33) and BDEC12), (22), (29), and BDEC5, we have $\neg \mathit{HoldsAt}(\mathit{Spilled}, S_1)$

From $\neg \mathit{Happens}(\mathit{LiftLeft}, S_0, S_2)$ (which follows from (33), (26), and (28)), and (30), we have $\mathit{Initiates}(\mathit{LiftRight}, \mathit{Spilled}, S_0, S_2)$. From this, $\mathit{Happens}(\mathit{LiftRight}, S_0, S_2)$ (which follows from (33)), and BDEC8, we have $\mathit{HoldsAt}(\mathit{Spilled}, S_2)$

From (30), (33), (26), (27), and (28), we have $\neg \exists e \, (\mathit{Happens}(e, S_0, S_2) \ \wedge \ \mathit{Initiates}(e, \mathit{Raised}, S_0, S_2))$. From this, $S(S_0, S_2)$ (which follows from (33) and BDEC12), (21), (29), and BDEC5, we have $\neg \mathit{HoldsAt}(\mathit{Raised}, S_2)$. ☐

Reactive Cat Scenario Consider a cat that eats food whenever food is present. If food is present at situation s_1, then the food is eaten between s_1 and every successor situation s_2 of s_1:

$$\begin{gathered}S(s_1, s_2) \wedge \mathit{HoldsAt}(\mathit{FoodPresent}, s_1) \rightarrow \\ \mathit{Happens}(\mathit{EatFood}, s_1, s_2)\end{gathered} \tag{34}$$

If the food is eaten, then it will no longer be present:

$$\mathit{Terminates}(\mathit{EatFood}, \mathit{FoodPresent}, s_1, s_2) \tag{35}$$

Food is initially present:

$$\mathit{HoldsAt}(\mathit{FoodPresent}, S_0) \tag{36}$$

The initial situation has one successor situation:

$$S(S_0, S_1) \tag{37}$$

We can then show that the cat will eat the food and that the food will no longer be present afterward.

Proposition 3. Let $\Sigma = (35)$, $\Delta = (34)$, $\Upsilon = (37)$, and $\Gamma = (36)$. Then we have

$$\begin{gathered}\mathit{CIRC}[\Sigma; \mathit{Initiates}, \mathit{Terminates}, \mathit{Releases}] \wedge \\ \mathit{CIRC}[\Delta; \mathit{Happens}] \wedge \Upsilon \wedge \Gamma \wedge \text{BDEC} \\ \vdash \mathit{Happens}(\mathit{EatFood}, S_0, S_1) \wedge \\ \neg \mathit{HoldsAt}(\mathit{FoodPresent}, S_1).\end{gathered}$$

Proof. From $\mathit{CIRC}[\Sigma; \mathit{Initiates}, \mathit{Terminates}, \mathit{Releases}]$ and Propositions 2 and 14 of Lifschitz (1994), we have

$$\neg \mathit{Initiates}(e, f, s_1, s_2) \tag{38}$$
$$\begin{gathered}\mathit{Terminates}(e, f, s_1, s_2) \leftrightarrow \\ (e = \mathit{EatFood} \wedge f = \mathit{FoodPresent})\end{gathered} \tag{39}$$
$$\neg \mathit{Releases}(e, f, s_1, s_2) \tag{40}$$

From $\mathit{CIRC}[\Delta; \mathit{Happens}]$ and Proposition 2 of Lifschitz, we have

$$\begin{gathered}\mathit{Happens}(e, s_1, s_2) \leftrightarrow \\ (e = \mathit{EatFood} \wedge S(s_1, s_2) \wedge \mathit{HoldsAt}(\mathit{FoodPresent}, s_1))\end{gathered} \tag{41}$$

From (36), (37), and (41), we have $\mathit{Happens}(\mathit{EatFood}, S_0, S_1)$. From this, $\mathit{Terminates}(\mathit{EatFood}, \mathit{FoodPresent}, S_0, S_1)$ (which follows from (39)), and BDEC9, we have $\neg \mathit{HoldsAt}(\mathit{FoodPresent}, S_1)$. ☐

Actual Situations and Events

In this section, we extend BDEC to distinguish between hypothetical and actual situations, and to distinguish between hypothetical and actual event occurrences, along the lines of Pinto and Reiter (1995).[3] We add the predicate $\mathit{Actual}(s)$, which represents that s is an actual situation, and the predicate $\mathit{ActuallyHappens}(e, s)$, which represents that event e actually occurs in situation s. We add several axioms. If

[3]See also the proposal of Baral, Gelfond, and Provetti (1997) in which action language $\mathcal{A}$ is modified to make an explicit distinction between hypothetical and actual actions.

a situation is actual, then a predecessor of that situation is actual:

$$\mathit{Actual}(s_2) \wedge S(s_1, s_2) \rightarrow \mathit{Actual}(s_1) \quad (42)$$

A situation has at most one actual successor:

$$S(s, s_1) \wedge S(s, s_2) \wedge \mathit{Actual}(s_1) \wedge \mathit{Actual}(s_2) \rightarrow s_1 = s_2 \quad (43)$$

An event actually occurs in a situation if and only if the event occurs between the situation and some actual situation:

$$\mathit{ActuallyHappens}(e, s_1) \leftrightarrow \exists s_2\, (\mathit{Happens}(e, s_1, s_2) \wedge \mathit{Actual}(s_2)) \quad (44)$$

Consider again the hypothetical Yale shooting scenario. We need only specify that S_1 is an actual situation:

$$\mathit{Actual}(S_1) \quad (45)$$

From this, $S(S_0, S_1)$ (which follows from (16) and BDEC12), and (42), we have $\mathit{Actual}(S_0)$. From $S(S_0, S_1)$, $S(S_0, S_2)$ (which follows from (16) and BDEC12), (45), and (43), we have $\neg\mathit{Actual}(S_2)$. From this, (16), and (44), we have $\neg\mathit{ActuallyHappens}(\mathit{Wait}, S_0)$. From $\mathit{Happens}(\mathit{Shoot}, S_0, S_1)$ (which follows from (16)), (45), and (44), we have $\mathit{ActuallyHappens}(\mathit{Shoot}, S_0)$.

Related Work

The main difference between BDEC and other proposals for combining the event calculus and situation calculus is that BDEC considers situations and timepoints to be one and the same. Other proposals define situations differently from timepoints.

Provetti (1996) proposes a hybrid of the event calculus and the situation calculus. Every timepoint is associated with a situation, but not every situation is associated with a timepoint. Every timepoint in the linear timeline of the event calculus corresponds to the root of a tree of hypothetical situations in the situation calculus. Unlike BDEC, the formalism does not handle hypothetical reasoning about concurrent actions.

Kowalski and Sadri (1997) propose a version of the event calculus with branching time and situations. Unlike BDEC, this variant disallows concurrent events, and it identifies timepoints with both situations and transitions between situations.

Van Belleghem, Denecker, and De Schreye (1997) propose a formalism that extends both the event calculus and the situation calculus. They start with a version of the event calculus with the linear time axiom $T_1 < T_2 \vee T_1 = T_2 \vee T_2 < T_1$. They then replace this axiom with the branching time axiom $(T_1 < T_3 \wedge T_2 < T_3) \rightarrow (T_1 < T_2 \vee T_1 = T_2 \vee T_2 < T_1)$. They then define a situation started by an event at timepoint T_1 as the set of timepoints T_2 after T_1 such that there are no events between T_1 and T_2. Unlike BDEC, this formalism disallows concurrent events, and defines situations as sets of timepoints.

Lévy and Quantz (1998) propose an extension of the event calculus in which a situation argument is added to every event calculus predicate, and the event calculus axioms are modified accordingly. This formalism does not have a successor relation between situations. Situations are instead related to one another by a predicate $\mathit{Equal\_until}(s_1, s_2, t)$, which represents that situations s_1 and s_2 are equal until time t.

The primary difference between BDEC and the situation calculus is that BDEC separates the *do* function into the two predicates *S* and *Happens*. BDEC is different from the formulation of the situation calculus of Reiter and colleagues (Lin & Reiter 1994; Pinto 1994; Pirri & Reiter 1999; Reiter 1993; 2001). Pirri and Reiter (1999) provide the following foundational axioms for the situation calculus:

$$\mathit{do}(a_1, s_1) = \mathit{do}(a_2, s_2) \supset a_1 = a_2 \wedge s_1 = s_2 \quad (46)$$

$$(\forall P).P(S_0) \wedge (\forall a, s)[P(s) \supset P(\mathit{do}(a, s))] \supset (\forall s)P(s) \quad (47)$$

$$\neg(s \sqsubset S_0) \quad (48)$$

$$s \sqsubset \mathit{do}(a, s') \equiv s = s' \vee s \sqsubset s' \quad (49)$$

In any model of these axioms, every situation has a successor for every element of the action domain. In any model of BDEC, if an event occurs between situation s_1 and s_2, then s_1 has s_2 as a successor (see axiom BDEC12).

Whereas BDEC allows more than one event between two situations, the situation calculus of Reiter et al. does not. In BDEC, we can write $\mathit{Happens}(E_1, S_0, S_1) \wedge \mathit{Happens}(E_2, S_0, S_1) \wedge E_1 \neq E_2$. The analogous situation calculus formula $S_1 = \mathit{do}(E_1, S_0) \wedge S_1 = \mathit{do}(E_2, S_0) \wedge E_1 \neq E_2$ is inconsistent with axiom (46). Multiple events between situations are also ruled out by Davis's (1994) axiom $\mathit{result}(\mathit{S1}, \mathit{EA}, \mathit{S2}) \wedge \mathit{result}(\mathit{S1}, \mathit{EB}, \mathit{S2}) \rightarrow \mathit{EA} = \mathit{EB}$.

Based on previous proposals, Reiter (1996) adds concurrent actions to the situation calculus by defining a concurrent action as a set of simple actions, and redefining *do* to work on concurrent actions. This enables representation of multiple events between situations. One can write $S_1 = \mathit{do}(\{A_1, A_2\}, S_0)$. Whereas BDEC allows zero events between two situations, Reiter requires a concurrent action to contain at least one simple action. (Zero actions between situations can be simulated in the situation calculus using an action that has no effects.)

BDEC is different from McCarthy's (1997; 2002) formulation of the situation calculus with concurrent events and narratives. In this formulation, the predicate $\mathit{Occurs}(e, s)$ represents that event e occurs in situation s, the function $\mathit{Next}(s)$ represents the next situation after s, and the function $\mathit{Result}(e, s)$ represents the situation that results from e occurring in situation s. These are related via the axiom

$$\mathit{Occurs}(e, s) \rightarrow \mathit{Next}(s) = \mathit{Result}(e, s) \quad (50)$$

We can represent that two events occur in S_0: $\mathit{Occurs}(E_1, S_0) \wedge \mathit{Occurs}(E_2, S_0) \wedge E_1 \neq E_2$. From this and (50), we have $\mathit{Next}(S_0) = \mathit{Result}(E_1, S_0)$ and $\mathit{Next}(S_0) = \mathit{Result}(E_2, S_0)$. Thus as in BDEC, we can have distinct events E_1 and E_2 between two situations:

$$S_1 = \mathit{Result}(E_1, S_0) = \mathit{Result}(E_2, S_0) \quad (51)$$

But we cannot then represent hypothetical events E_1 and E_3 that occur between S_0 and some other situation S_2:

$$S_2 = \mathit{Result}(E_1, S_0) \quad (52)$$

$$S_2 = \mathit{Result}(E_3, S_0) \quad (53)$$

$$S_1 \neq S_2 \quad (54)$$

The formulas (52) and (54) are inconsistent with (51). To allow hypothetical reasoning, McCarthy (1997) adds a narrative argument to *Occurs*, and uses contexts (Guha 1992; McCarthy 1993) to reason with $\mathit{Occurs}(e,s)$ inside a narrative. We rewrite (50) as $\mathit{Occurs}(e,s,n) \rightarrow \mathit{Next}(s,n) = \mathit{Result}(e,s,n)$ and use lifting formulas such as $\mathit{SpecializeNarrative}(n,c',c) \wedge \mathit{Ist}(c',\mathit{Occurs}(e,s)) \rightarrow \mathit{Ist}(c,\mathit{Occurs}(e,s,n))$. Nossum and Thielscher (1999) propose to use contexts in the event calculus.

Conclusions

We introduced BDEC, a branching time discrete version of the classical logic event calculus. BDEC is useful for commonsense reasoning about hypothetical events as well as other phenomena of action and change. A catalog of commonsense phenomena treated by the classical logic event calculus is provided by Mueller (2006a).

Some areas for further work are the following:

- BDEC allows two identical sets of event occurrences to lead to two different situations. For example, E_1 and E_2 could lead from S_0 to S_1, and E_1 and E_2 could also lead from S_0 to a situation S_2 distinct from S_1. The following axiom to rule this out could be added: $(\forall e\,(\mathit{Happens}(e,s,s_1) \leftrightarrow \mathit{Happens}(e,s,s_2))) \rightarrow s_1 = s_2$.
- To allow reasoning about gradual change, the *Trajectory* and *AntiTrajectory* predicates (Miller & Shanahan 2002) could be added to BDEC. This requires definition of the distance between two situations along a path.

References

Baral, C.; Gelfond, M.; and Provetti, A. 1997. Representing actions. *Journal of Logic Programming* 31(1–3):201–243.

Davis, E. 1994. Knowledge preconditions for plans. *Journal of Logic and Computation* 4(5):721–766.

Doherty, P.; Gustafsson, J.; Karlsson, L.; and Kvarnström, J. 1998. TAL. *Linköping Electronic Articles in Computer and Information Science* 3(015).

Gelfond, M.; Lifschitz, V.; and Rabinov, A. 1991. What are the limitations of the situation calculus? In *Automated Reasoning*. Kluwer. 167–179.

Guha, R. V. 1992. *Contexts*. Ph.D. Dissertation, Stanford University.

Hanks, S., and McDermott, D. V. 1987. Nonmonotonic logic and temporal projection. *Artificial Intelligence* 33(3):379–412.

Kowalski, R. A., and Sadri, F. 1997. Reconciling the event calculus with the situation calculus. *Journal of Logic Programming* 31(1–3):39–58.

Lévy, F., and Quantz, J. J. 1998. Representing beliefs in a situated event calculus. In *Proc. ECAI 1998*, 547–551.

Lifschitz, V. 1994. Circumscription. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3. Oxford University Press. 298–352.

Lin, F., and Reiter, R. 1994. State constraints revisited. *Journal of Logic and Computation* 4(5):655–678.

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

McCarthy, J. 1980. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence* 13(1–2):27–39.

McCarthy, J. 1993. Notes on formalizing context. In *Proc. IJCAI 1993*, 555–562.

McCarthy, J. 1997. Situation calculus with concurrent events and narrative. `http://www-formal.stanford.edu/jmc/narrative/narrative.html`.

McCarthy, J. 2002. Actions and other events in situation calculus. In *Proc. KR 2002*, 615–626.

Miller, R., and Shanahan, M. 2002. Some alternative formulations of the event calculus. In *Computational Logic*, volume 2408 of *LNCS*. Springer. 452–490.

Morgenstern, L. 2001. Mid-sized axiomatizations of commonsense problems. *Studia Logica* 67:333–384.

Mueller, E. T. 2004a. Event calculus reasoning through satisfiability. *Journal of Logic and Computation* 14(5):703–730.

Mueller, E. T. 2004b. A tool for satisfiability-based commonsense reasoning in the event calculus. In *Proc. FLAIRS 2004*, 147–152.

Mueller, E. T. 2006a. *Commonsense Reasoning*. Morgan Kaufmann.

Mueller, E. T. 2006b. Event calculus and temporal action logics compared. *Artificial Intelligence* 170(11):1017–1029.

Mueller, E. T. in submission. Discrete event calculus with branching time (extended version). `http://www.signiform.com/erik/pubs/ecsc.pdf`.

Nossum, R., and Thielscher, M. 1999. Counterfactual reasoning by means of a calculus of narrative context. In *Modeling and Using Context*, volume 1688 of *LNCS*. Springer. 495–498.

Pinto, J. A., and Reiter, R. 1995. Reasoning about time in the situation calculus. *Annals of Mathematics and Artificial Intelligence* 14(2–4):251–268.

Pinto, J. A. 1994. *Temporal Reasoning in the Situation Calculus*. Ph.D. Dissertation, University of Toronto.

Pirri, F., and Reiter, R. 1999. Some contributions to the metatheory of the situation calculus. *Journal of the ACM* 46(3):325–361.

Provetti, A. 1996. Hypothetical reasoning about actions. *Computational Intelligence* 12:478–498.

Reiter, R. 1993. Proving properties of states in the situation calculus. *Artificial Intelligence* 64(2):337–351.

Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In *Proc. KR 1996*, 2–13.

Reiter, R. 2001. *Knowledge in Action*. MIT Press.

Schmidt, J. 1960. Peano-Bäume. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 6(3/4):225–239.

Shanahan, M. 1996. Robotics and the common sense informatic situation. In *Proc. ECAI 1996*, 684–688.

Shanahan, M. 1997. *Solving the Frame Problem*. MIT Press.

Shanahan, M. 1999a. The event calculus explained. In *Artificial Intelligence Today*, volume 1600 of *LNCS*. Springer. 409–430.

Shanahan, M. 1999b. The ramification problem in the event calculus. In *Proc. IJCAI 1999*, 140–146.

Shanahan, M. 2004. An attempt to formalise a non-trivial benchmark problem in common sense reasoning. *Artificial Intelligence* 153:141–165.

Van Belleghem, K.; Denecker, M.; and De Schreye, D. 1997. On the relation between situation calculus and event calculus. *Journal of Logic Programming* 31(1–3):3–37.