Discrete event calculus with branching time

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

Preprint of December 21, 2006

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 characterize the precise relationship between the linear and branching versions of the discrete event calculus, and prove that a restricted version of the branching discrete event calculus is equivalent to the situation 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.

1 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, 1963; 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, Gustafsson, Karlsson, & Kvarnström, 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 characterize the precise relationship between LDEC and BDEC, and prove that a restricted version of BDEC is equivalent to the situation calculus. 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.

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.

2 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:

- $\mathit{Happens}_L(e, s)$: Event e occurs at situation s.
- $\mathit{HoldsAt}(f, s)$: Fluent f is true at situation s.
- $\mathit{ReleasedAt}(f, s)$: Fluent f is released from the commonsense law of inertia at situation s.
- $\mathit{Initiates}_L(e, f, s)$: 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.
- $\mathit{Terminates}_L(e, f, s)$: 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.
- $\mathit{Releases}_L(e, f, s)$: 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 (Lifschitz, 1987; Shanahan, 1997) states that a fluent's truth value persists unless the fluent is affected by an event. When a fluent is released from this law, its truth value can fluctuate. Fluents that are released from the commonsense law of

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

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.

3 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 Section 6.2).

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.

- *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) and Clark and Tärnlund (1977), 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]

4 Relationship of LDEC and BDEC

The relationship of LDEC and BDEC can be characterized precisely as follows. Let L be the conjunction of the following axioms:

L1. $S(s_1, s_2) \leftrightarrow S_L(s_1) = s_2$

L2. $\mathit{Happens}(e, s_1, s_2) \leftrightarrow \mathit{Happens}_L(e, s_1) \wedge S_L(s_1) = s_2$

L3. $\mathit{Initiates}(e, f, s_1, s_2) \leftrightarrow \mathit{Initiates}_L(e, f, s_1)$

L4. $\mathit{Terminates}(e, f, s_1, s_2) \leftrightarrow \mathit{Terminates}_L(e, f, s_1)$

L5. $\mathit{Releases}(e, f, s_1, s_2) \leftrightarrow \mathit{Releases}_L(e, f, s_1)$

[2]See the proof of Proposition 2.1 of Pinto (1994, pp. 103–105) and Exercise 2.8.3 of Manzano (1996, p. 90).

L serves as a bridge between BDEC and LDEC. We can show that the conjunction of BDEC and L is logically equivalent to the conjunction of LDEC and L.

We first prove several lemmas.

Lemma 1. LDEC $\wedge$ L $\vdash$ BDEC8.

Proof. Suppose $LDEC \wedge L$. Let e be an arbitrary event, f be an arbitrary fluent, and s_1 and s_2 be arbitrary situations. We must show $Happens(e, s_1, s_2) \wedge Initiates(e, f, s_1, s_2) \rightarrow HoldsAt(f, s_2)$. Suppose

$$Happens(e, s_1, s_2) \tag{1}$$

$$Initiates(e, f, s_1, s_2) \tag{2}$$

From (1) and L2, we have

$$Happens_L(e, s_1) \wedge S_L(s_1) = s_2 \tag{3}$$

From (2) and L3, we have $Initiates_L(e, f, s_1)$. From this, (3), and LDEC8, we have $HoldsAt(f, s_2)$, as required. □

Lemma 2. LDEC $\wedge$ L $\vdash$ BDEC9.

Proof. The proof is identical to that of Lemma 1, except that $\neg HoldsAt$ is substituted for *HoldsAt*, *Terminates* is substituted for *Initiates*, $Terminates_L$ is substituted for $Initiates_L$, and LDEC9 is substituted for LDEC8. □

Lemma 3. LDEC $\wedge$ L $\vdash$ BDEC10.

Proof. The proof is identical to that of Lemma 1, except that *ReleasedAt* is substituted for *HoldsAt*, *Releases* is substituted for *Initiates*, $Releases_L$ is substituted for $Initiates_L$, and LDEC10 is substituted for LDEC8. □

Lemma 4. LDEC $\wedge$ L $\vdash$ BDEC11.

Proof. Suppose $LDEC \wedge L$. Let e be an arbitrary event, f be an arbitrary fluent, and s_1 and s_2 be arbitrary situations. We must show $Happens(e, s_1, s_2) \wedge (Initiates(e, f, s_1, s_2) \vee Terminates(e, f, s_1, s_2)) \rightarrow \neg ReleasedAt(f, s_2)$. Suppose

$$Happens(e, s_1, s_2) \tag{4}$$

$$Initiates(e, f, s_1, s_2) \vee Terminates(e, f, s_1, s_2) \tag{5}$$

From (4) and L2, we have

$$Happens_L(e, s_1) \wedge S(s_1) = s_2 \tag{6}$$

From (5), L3, and L4, we have $Initiates_L(e, f, s_1) \vee Terminates_L(e, f, s_1)$. From this, (6), and LDEC11, we have $\neg ReleasedAt(f, s_2)$, as required. □

Now we proceed to the equivalence results.

Theorem 1. BDEC $\wedge$ L $\vdash$ LDEC.

Proof. Suppose *BDEC* $\wedge$ *L*.
LDEC1 follows from BDEC1 and L1.
LDEC2 follows from BDEC2 and L1.
LDEC3 follows from BDEC3 and L1.
LDEC4 follows from BDEC4, L1, L2, and L4.
LDEC5 follows from BDEC5, L1, L2, and L3.
LDEC6 follows from BDEC6, L1, L2, L3, and L4.
LDEC7 follows from BDEC7, L1, L2, and L5.
LDEC8 follows from BDEC8, L2, and L3.
LDEC9 follows from BDEC9, L2, and L4.
LDEC10 follows from BDEC10, L2, and L5.
LDEC11 follows from BDEC11, L2, L3, and L4.
Therefore, *LDEC*. □

Theorem 2. LDEC $\wedge$ L $\vdash$ BDEC.

Proof. Suppose *LDEC* $\wedge$ *L*.
BDEC1 follows from LDEC1 and L1.
BDEC2 follows from LDEC2 and L1.
BDEC3 follows from LDEC3 and L1.
BDEC4 follows from LDEC4, L1, L2, and L4.
BDEC5 follows from LDEC5, L1, L2, and L3.
BDEC6 follows from LDEC6, L1, L2, L3, and L4.
BDEC7 follows from LDEC7, L1, L2, and L5.
BDEC8 follows from Lemma 1.
BDEC9 follows from Lemma 2.
BDEC10 follows from Lemma 3.
BDEC11 follows from Lemma 4.
BDEC12 follows from L1 and L2.
Therefore, *BDEC*. □

Corollary 1. BDEC $\wedge$ L *is logically equivalent to* LDEC $\wedge$ L*.*

Proof. This follows from Theorem 1 and Theorem 2. □

5 Relationship of BDEC and situation calculus

If BDEC is restricted appropriately, we can show that it is equivalent to the situation calculus. We restrict BDEC as follows:

1. We eliminate the ability to release fluents from the commonsense law of inertia.
2. We require a unique successor of an event.
3. We disallow multiple events between situations.

4. We require *Initiates* and *Terminates* to be independent of the successor.

Let BDECS be the conjunction of the following axioms:

BDECS1. $\neg S(s, S_0)$
BDECS2. $S(s_1, s) \wedge S(s_2, s) \rightarrow s_1 = s_2$
BDECS3. $\forall P\, ((P(S_0) \wedge \forall s_1, s_2\, (S(s_1, s_2) \wedge P(s_1) \rightarrow P(s_2))) \rightarrow \forall s\, P(s))$
BDECS4. $S(s_1, s_2) \wedge \mathit{HoldsAt}(f, s_1) \wedge$
$\neg\exists e\, (\mathit{Happens}(e, s_1, s_2) \wedge \mathit{Terminates}(e, f, s_1, s_2)) \rightarrow$
$\mathit{HoldsAt}(f, s_2)$
BDECS5. $S(s_1, s_2) \wedge \neg\mathit{HoldsAt}(f, s_1) \wedge$
$\neg\exists e\, (\mathit{Happens}(e, s_1, s_2) \wedge \mathit{Initiates}(e, f, s_1, s_2)) \rightarrow$
$\neg\mathit{HoldsAt}(f, s_2)$
BDECS6. $\mathit{Happens}(e, s_1, s_2) \wedge \mathit{Initiates}(e, f, s_1, s_2) \rightarrow \mathit{HoldsAt}(f, s_2)$
BDECS7. $\mathit{Happens}(e, s_1, s_2) \wedge \mathit{Terminates}(e, f, s_1, s_2) \rightarrow \neg\mathit{HoldsAt}(f, s_2)$
BDECS8. $\mathit{Happens}(e, s_1, s_2) \rightarrow S(s_1, s_2)$
BDECS9. $\mathit{Happens}(e, s, s_1) \wedge \mathit{Happens}(e, s, s_2) \rightarrow s_1 = s_2$
BDECS10. $\exists s_2\, \mathit{Happens}(e, s_1, s_2)$
BDECS11. $\mathit{Happens}(e_1, s_1, s_2) \wedge \mathit{Happens}(e_2, s_1, s_2) \rightarrow e_1 = e_2$
BDECS12. $\mathit{Initiates}(e, f, s, s_1) \leftrightarrow \mathit{Initiates}(e, f, s, s_2)$
BDECS13. $\mathit{Terminates}(e, f, s, s_1) \leftrightarrow \mathit{Terminates}(e, f, s, s_2)$

Axioms BDECS1 through BDECS8 are obtained by removing mention of the predicates *ReleasedAt* and *Releases* from BDEC. Axioms BDECS9 and BDECS10 require a unique successor of an event. Axiom BDECS11 disallows multiple events between situations. Axioms BDECS12 and BDECS13 require *Initiates* and *Terminates* to be independent of the successor.

We use a version of the situation calculus similar to that of Kowalski and Sadri (1997). Let SC be the conjunction of the following axioms:

SC1. $\mathit{Do}(e, s) \neq S_0$
SC2. $\mathit{Do}(e_1, s_1) = \mathit{Do}(e_2, s_2) \rightarrow e_1 = e_2 \wedge s_1 = s_2$
SC3. $\forall P\, ((P(S_0) \wedge \forall e, s\, (P(s) \rightarrow P(\mathit{Do}(e, s)))) \rightarrow \forall s\, P(s))$
SC4. $\mathit{HoldsAt}(f, s) \wedge \neg\mathit{Terminates}_S(e, f, s) \rightarrow \mathit{HoldsAt}(f, \mathit{Do}(e, s))$
SC5. $\neg\mathit{HoldsAt}(f, s) \wedge \neg\mathit{Initiates}_S(e, f, s) \rightarrow \neg\mathit{HoldsAt}(f, \mathit{Do}(e, s))$
SC6. $\mathit{Initiates}_S(e, f, s) \rightarrow \mathit{HoldsAt}(f, \mathit{Do}(e, s))$
SC7. $\mathit{Terminates}_S(e, f, s) \rightarrow \neg\mathit{HoldsAt}(f, \mathit{Do}(e, s))$

Let S be the conjunction of the following axioms:

S1. $S(s_1, s_2) \leftrightarrow \exists e\, (\mathit{Do}(e, s_1) = s_2)$
S2. $\mathit{Happens}(e, s_1, s_2) \leftrightarrow \mathit{Do}(e, s_1) = s_2$
S3. $\mathit{Initiates}(e, f, s_1, s_2) \leftrightarrow \mathit{Initiates}_S(e, f, s_1)$
S4. $\mathit{Terminates}(e, f, s_1, s_2) \leftrightarrow \mathit{Terminates}_S(e, f, s_1)$

S serves as a bridge between BDECS and S. We can show that the conjunction of BDECS and S is logically equivalent to the conjunction of SC and S.

We start by proving a number of lemmas.

Lemma 5. BDECS $\wedge$ S $\vdash$ SC2.

Proof. Suppose $BDECS \wedge S$. Let e_1 and e_2 be arbitrary events and s_1 and s_2 be arbitrary situations. We must show $Do(e_1, s_1) = Do(e_2, s_2) \rightarrow e_1 = e_2 \wedge s_1 = s_2$. Suppose

$$Do(e_1, s_1) = Do(e_2, s_2) \tag{7}$$

Let $s_3 = Do(e_1, s_1)$. From this and S2, we have

$$Happens(e_1, s_1, s_3) \tag{8}$$

From $s_3 = Do(e_1, s_1)$, (7), and S2, we have

$$Happens(e_2, s_2, s_3) \tag{9}$$

From (8) and BDECS8, we have

$$S(s_1, s_3) \tag{10}$$

From (9) and BDECS8, we have

$$S(s_2, s_3) \tag{11}$$

From this, (10), and BDECS2, we have $s_1 = s_2$, as required. From this, (8), (9), and BDECS11, we have $e_1 = e_2$, as required. □

Lemma 6. BDECS $\wedge$ S $\vdash$ SC3.

Proof. Suppose $BDECS \wedge S$. Let P be an arbitrary predicate. We must show $(P(S_0) \wedge \forall e, s\,(P(s) \rightarrow P(Do(e, s)))) \rightarrow \forall s\, P(s)$. Suppose

$$P(S_0) \tag{12}$$
$$\forall e, s\,(P(s) \rightarrow P(Do(e, s))) \tag{13}$$

We can show

$$\forall s_1, s_2\,(S(s_1, s_2) \wedge P(s_1) \rightarrow P(s_2)) \tag{14}$$

To see this, let s_1 and s_2 be arbitrary situations. Suppose

$$S(s_1, s_2) \tag{15}$$
$$P(s_1) \tag{16}$$

From (15) and S1, we have $Do(E, s_1) = s_2$ for some E. From this, (16), and (13), we have $P(s_2)$, as required.

From (12), (14), and BDECS3, we have $\forall s\, P(s)$, as required. □

Lemma 7. BDECS $\wedge$ S $\vdash$ SC4.

Proof. Suppose $BDECS \wedge S$. Let e be an arbitrary event, f be an arbitrary fluent, and s_1 be an arbitrary situation. We must show $HoldsAt(f, s_1) \wedge \neg Terminates_S(e, f, s_1) \rightarrow HoldsAt(f, Do(e, s_1))$. Suppose

$$HoldsAt(f, s_1) \tag{17}$$

$$\neg Terminates_S(e, f, s_1) \tag{18}$$

Let $s_2 = Do(e, s_1)$. From this and S1, we have

$$S(s_1, s_2) \tag{19}$$

From $s_2 = Do(e, s_1)$ and S2, we have $Happens(e, s_1, s_2)$. From this and BDECS11, we have

$$Happens(e', s_1, s_2) \rightarrow e' = e \tag{20}$$

From (18) and S4, we have $\neg Terminates(e, f, s_1, s_2)$. From this and (20), we have $\neg \exists e' \, (Happens(e', s_1, s_2) \wedge Terminates(e', f, s_1, s_2))$. From this, (19), (17), and BDECS4, we have $HoldsAt(f, s_2)$. From this and $s_2 = Do(e, s_1)$, we have $HoldsAt(f, Do(e, s_1))$, as required. □

Lemma 8. BDECS ∧ S ⊢ SC5.

Proof. The proof is identical to that of Lemma 7, except that $\neg HoldsAt$ is substituted for $HoldsAt$, $Initiates_S$ is substituted for $Terminates_S$, $Initiates$ is substituted for $Terminates$, and BDECS5 is substituted for BDECS4. □

Lemma 9. BDECS ∧ S ⊢ SC6.

Proof. Suppose $BDECS \wedge S$. Let e be an arbitrary event, f be an arbitrary fluent, and s_1 be an arbitrary situation. We must show $Initiates_S(e, f, s_1) \rightarrow HoldsAt(f, Do(e, s_1))$. Suppose

$$Initiates_S(e, f, s_1) \tag{21}$$

Let $s_2 = Do(e, s_1)$. From this and S2, we have

$$Happens(e, s_1, s_2) \tag{22}$$

From (21) and S3, we have $Initiates(e, f, s_1, s_2)$. From this, (22), and BDECS6, we have $HoldsAt(f, s_2)$. From this and $s_2 = Do(e, s_1)$, we have $HoldsAt(f, Do(e, s_1))$, as required. □

Lemma 10. BDECS ∧ S ⊢ SC7.

Proof. The proof is identical to that of Lemma 9, except that $\neg HoldsAt$ is substituted for $HoldsAt$, $Terminates_S$ is substituted for $Initiates_S$, $Terminates$ is substituted for $Initiates$, and BDECS7 is substituted for BDECS6. □

Lemma 11. SC ∧ S ⊢ BDECS2.

Proof. Suppose $SC \wedge S$. Let s_1, s_2, and s be arbitrary situations. We must show $S(s_1, s) \wedge S(s_2, s) \rightarrow s_1 = s_2$. Suppose $S(s_1, s) \wedge S(s_2, s)$. From this and S1, we have $Do(E_1, s_1) = Do(E_2, s_2)$ for some E_1 and E_2. From this and SC2, we have $s_1 = s_2$, as required. □

Lemma 12. SC $\wedge$ S $\vdash$ BDECS3.

Proof. Suppose $SC \wedge S$. Let P be an arbitrary predicate. We must show $(P(S_0) \wedge \forall s_1, s_2\, (S(s_1, s_2) \wedge P(s_1) \rightarrow P(s_2))) \rightarrow \forall s\, P(s)$. Suppose

$$P(S_0) \tag{23}$$

$$\forall s_1, s_2\, (S(s_1, s_2) \wedge P(s_1) \rightarrow P(s_2)) \tag{24}$$

We can show

$$\forall e, s\, (P(s) \rightarrow P(Do(e, s))) \tag{25}$$

To see this, let e be an arbitrary event and s be an arbitrary situation. Suppose

$$P(s) \tag{26}$$

Let $s_3 = Do(e, s)$. From this and S1, we have $S(s, s_3)$. From this, (26), and (24), we have $P(s_3)$. From this and $s_3 = Do(e, s)$, we have $P(Do(e, s))$, as required.

From (23), (25), and SC3, we have $\forall s\, P(s)$, as required. ☐

Lemma 13. SC $\wedge$ S $\vdash$ BDECS4.

Proof. Suppose $SC \wedge S$. Let f be an arbitrary fluent and s_1 and s_2 be arbitrary situations. We must show $S(s_1, s_2) \wedge HoldsAt(f, s_1) \wedge \neg\exists e\, (Happens(e, s_1, s_2) \wedge Terminates(e, f, s_1, s_2)) \rightarrow HoldsAt(f, s_2)$. Suppose

$$S(s_1, s_2) \tag{27}$$

$$HoldsAt(f, s_1) \tag{28}$$

$$\neg\exists e\, (Happens(e, s_1, s_2) \wedge Terminates(e, f, s_1, s_2)) \tag{29}$$

From (29), we have

$$Happens(e, s_1, s_2) \rightarrow \neg Terminates(e, f, s_1, s_2) \tag{30}$$

From (27) and S1, we have

$$Do(E, s_1) = s_2 \tag{31}$$

for some E. From this and S2, we have $Happens(E, s_1, s_2)$. From this and (30), we have $\neg Terminates(E, f, s_1, s_2)$. From this and S4, we have $\neg Terminates_S(E, f, s_1)$. From this, (28), and SC4, we have $HoldsAt(f, Do(E, s_1))$. From this and (31), we have $HoldsAt(f, s_2)$, as required. ☐

Lemma 14. SC $\wedge$ S $\vdash$ BDECS5.

Proof. The proof is identical to that of Lemma 13, except that $\neg HoldsAt$ is substituted for $HoldsAt$, $Initiates$ is substituted for $Terminates$, $Initiates_S$ is substituted for $Terminates_S$, and SC5 is substituted for SC4. ☐

Lemma 15. SC $\wedge$ S $\vdash$ BDECS6.

Proof. Suppose $SC \wedge S$. Let e be an arbitrary event, f be an arbitrary fluent, and s_1 and s_2 be arbitrary situations. We must show $Happens(e, s_1, s_2) \wedge Initiates(e, f, s_1, s_2) \rightarrow HoldsAt(f, s_2)$. Suppose

$$Happens(e, s_1, s_2) \tag{32}$$
$$Initiates(e, f, s_1, s_2) \tag{33}$$

From (33) and S3, we have

$$Initiates_S(e, f, s_1) \tag{34}$$

From (32) and S2, we have $Do(e, s_1) = s_2$. From this, (34), and SC6, we have $HoldsAt(f, s_2)$, as required. □

Lemma 16. SC $\wedge$ S $\vdash$ BDECS7.

Proof. The proof is identical to that of Lemma 15, except that $\neg HoldsAt$ is substituted for $HoldsAt$, $Terminates$ is substituted for $Initiates$, $Terminates_S$ is substituted for $Initiates_S$, and SC7 is substituted for SC6. □

Lemma 17. SC $\wedge$ S $\vdash$ BDECS9.

Proof. Suppose $SC \wedge S$. Let e be an arbitrary event and s, s_1, and s_2 be arbitrary situations. We must show $Happens(e, s, s_1) \wedge Happens(e, s, s_2) \rightarrow s_1 = s_2$. Suppose

$$Happens(e, s, s_1) \tag{35}$$
$$Happens(e, s, s_2) \tag{36}$$

From (35) and S2, we have

$$Do(e, s) = s_1 \tag{37}$$

From (36) and S2, we have $Do(e, s) = s_2$. From this and (37), we have $s_1 = s_2$, as required. □

Lemma 18. SC $\wedge$ S $\vdash$ BDECS10.

Proof. Suppose $SC \wedge S$. Let e be an arbitrary event and s_1 be an arbitrary situation. We must show $Happens(e, s_1, s_2)$ for some situation s_2. Let $s_2 = Do(e, s_1)$. From S2, we have $Happens(e, s_1, s_2)$, as required. □

Now we proceed to the equivalence results.

Theorem 3. BDECS $\wedge$ S $\vdash$ SC.

Proof. Suppose $BDECS \wedge S$.
SC1 follows from BDECS1 and S1.
SC2 follows from Lemma 5.
SC3 follows from Lemma 6.
SC4 follows from Lemma 7.
SC5 follows from Lemma 8.
SC6 follows from Lemma 9.
SC7 follows from Lemma 10.
Therefore, *SC*. □

Theorem 4. $\mathrm{SC} \wedge \mathrm{S} \vdash \mathrm{BDECS}$.

Proof. Suppose $SC \wedge S$.
BDECS1 follows from SC1 and S1.
BDECS2 follows from Lemma 11.
BDECS3 follows from Lemma 12.
BDECS4 follows from Lemma 13.
BDECS5 follows from Lemma 14.
BDECS6 follows from Lemma 15.
BDECS7 follows from Lemma 16.
BDECS8 follows from S1 and S2.
BDECS9 follows from Lemma 17.
BDECS10 follows from Lemma 18.
BDECS11 follows from SC2 and S2.
BDECS12 follows from S3.
BDECS13 follows from S4.
Therefore, *BDECS*. □

Corollary 2. $\mathrm{BDECS} \wedge \mathrm{S}$ *is logically equivalent to* $\mathrm{SC} \wedge \mathrm{S}$.

Proof. This follows from Theorem 3 and Theorem 4. □

6 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.

6.1 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).[3]

[3]Gelfond and Lifschitz (1993) consider a related problem of reasoning about alternative futures based on the Yale shooting scenario.

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

$$Initiates(Load, Loaded, s_1, s_2) \tag{38}$$

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

$$\begin{aligned} &HoldsAt(Loaded, s_1) \rightarrow \\ &Terminates(Shoot, Alive, s_1, s_2) \end{aligned} \tag{39}$$

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

$$Terminates(Shoot, Loaded, s_1, s_2) \tag{40}$$

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

$$HoldsAt(Alive, S_0) \tag{41}$$

The gun is shot between situation S_0 and situation S_1:

$$Happens(Shoot, S_0, S_1) \tag{42}$$

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:

$$Happens(Wait, S_0, S_2) \tag{43}$$

$$HoldsAt(Loaded, S_2) \tag{44}$$

The events and fluents are distinct:

$$Load \neq Shoot \tag{45}$$

$$Shoot \neq Wait \tag{46}$$

$$Load \neq Wait \tag{47}$$

$$Alive \neq Loaded \tag{48}$$

Fluents are never released from the commonsense law of inertia:

$$\neg ReleasedAt(f, s) \tag{49}$$

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* $\Sigma = (38) \wedge (39) \wedge (40)$*,* $\Delta = (42) \wedge (43)$*,* $\Omega = (45) \wedge (46) \wedge (47) \wedge (48)$*, and* $\Gamma = (41) \wedge (44) \wedge (49)$*. Then we have*

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

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

$$\mathit{Initiates}(e, f, s_1, s_2) \leftrightarrow (e = \mathit{Load} \wedge f = \mathit{Loaded}) \tag{50}$$

$$\begin{gathered}\mathit{Terminates}(e, f, s_1, s_2) \leftrightarrow \\ (e = \mathit{Shoot} \wedge f = \mathit{Alive} \wedge \mathit{HoldsAt}(\mathit{Loaded}, s_1)) \ \vee \\ (e = \mathit{Shoot} \wedge f = \mathit{Loaded})\end{gathered} \tag{51}$$

$$\neg \mathit{Releases}(e, f, s_1, s_2) \tag{52}$$

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

$$\begin{gathered}\mathit{Happens}(e, s_1, s_2) \leftrightarrow \\ (e = \mathit{Shoot} \wedge s_1 = S_0 \wedge s_2 = S_1) \ \vee \\ (e = \mathit{Wait} \wedge s_1 = S_0 \wedge s_2 = S_2)\end{gathered} \tag{53}$$

Seeking a contradiction, suppose that

$$\neg \mathit{HoldsAt}(\mathit{Loaded}, S_0) \tag{54}$$

From (50), (53), (45), and (47), 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 (53) and BDEC12), (54), (49), and BDEC5, we have $\neg \mathit{HoldsAt}(\mathit{Loaded}, S_2)$, which contradicts (44). Therefore, $\mathit{HoldsAt}(\mathit{Loaded}, S_0)$.

From this and (51), 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 (53)), and BDEC9, we have $\neg \mathit{HoldsAt}(\mathit{Alive}, S_1)$. □

6.2 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:

$$\begin{gathered}\mathit{Happens}(\mathit{LiftLeft}, s_1, s_2) \rightarrow \\ \mathit{Initiates}(\mathit{LiftRight}, \mathit{Raised}, s_1, s_2)\end{gathered} \tag{55}$$

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

$$\neg Happens(LiftRight, s_1, s_2) \rightarrow \\ Initiates(LiftLeft, Spilled, s_1, s_2) \tag{56}$$

$$\neg Happens(LiftLeft, s_1, s_2) \rightarrow \\ Initiates(LiftRight, Spilled, s_1, s_2) \tag{57}$$

Initially, the bowl is not raised and not spilled:

$$\neg HoldsAt(Raised, S_0) \tag{58}$$

$$\neg HoldsAt(Spilled, S_0) \tag{59}$$

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

$$Happens(LiftLeft, S_0, S_1) \tag{60}$$

$$Happens(LiftRight, S_0, S_1) \tag{61}$$

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:

$$Happens(LiftRight, S_0, S_2) \tag{62}$$

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

The events and fluents are distinct:

$$LiftLeft \neq LiftRight \tag{63}$$

$$Raised \neq Spilled \tag{64}$$

Situations S_1 and S_2 are distinct:

$$S_1 \neq S_2 \tag{65}$$

Fluents are never released from the commonsense law of inertia:

$$\neg ReleasedAt(f, s) \tag{66}$$

Proposition 2. *Let* $\Sigma = (55) \wedge (56) \wedge (57)$*,* $\Delta = (60) \wedge (61) \wedge (62)$*,* $\Omega = (63) \wedge (64) \wedge (65)$*, and* $\Gamma = (58) \wedge (59) \wedge (66)$*. Then we have*

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

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

$$\begin{array}{c} \mathit{Initiates}(e, f, s_1, s_2) \leftrightarrow \\ (e = \mathit{LiftRight} \wedge f = \mathit{Raised} \wedge \\ \mathit{Happens}(\mathit{LiftLeft}, s_1, s_2)) \;\vee \\ (e = \mathit{LiftLeft} \wedge f = \mathit{Spilled} \wedge \\ \neg \mathit{Happens}(\mathit{LiftRight}, s_1, s_2)) \;\vee \\ (e = \mathit{LiftRight} \wedge f = \mathit{Spilled} \wedge \\ \neg \mathit{Happens}(\mathit{LiftLeft}, s_1, s_2)) \end{array} \tag{67}$$

$$\neg \mathit{Terminates}(e, f, s_1, s_2) \tag{68}$$

$$\neg \mathit{Releases}(e, f, s_1, s_2) \tag{69}$$

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

$$\begin{array}{c} \mathit{Happens}(e, s_1, s_2) \leftrightarrow \\ (e = \mathit{LiftLeft} \wedge s_1 = S_0 \wedge s_2 = S_1) \;\vee \\ (e = \mathit{LiftRight} \wedge s_1 = S_0 \wedge s_2 = S_1) \;\vee \\ (e = \mathit{LiftRight} \wedge s_1 = S_0 \wedge s_2 = S_2) \end{array} \tag{70}$$

From $\mathit{Happens}(\mathit{LiftLeft}, S_0, S_1)$ (which follows from (70)) and (67), 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 (70)), and BDEC8, we have $\mathit{HoldsAt}(\mathit{Raised}, S_1)$.

From (67), (70), (63), (64), and (65), 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 (70) and BDEC12), (59), (66), and BDEC5, we have $\neg \mathit{HoldsAt}(\mathit{Spilled}, S_1)$

From $\neg \mathit{Happens}(\mathit{LiftLeft}, S_0, S_2)$ (which follows from (70), (63), and (65)), and (67), 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 (70)), and BDEC8, we have $\mathit{HoldsAt}(\mathit{Spilled}, S_2)$

From (67), (70), (63), (64), and (65), 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 (70) and BDEC12), (58), (66), and BDEC5, we have $\neg \mathit{HoldsAt}(\mathit{Raised}, S_2)$. □

6.3 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:

$$\begin{array}{c} S(s_1, s_2) \wedge \mathit{HoldsAt}(\mathit{FoodPresent}, s_1) \rightarrow \\ \mathit{Happens}(\mathit{EatFood}, s_1, s_2) \end{array} \tag{71}$$

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

$$\mathit{Terminates}(\mathit{EatFood}, \mathit{FoodPresent}, s_1, s_2) \tag{72}$$

Food is initially present:

$$HoldsAt(FoodPresent, S_0) \tag{73}$$

The initial situation has one successor situation:

$$S(S_0, S_1) \tag{74}$$

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

Proposition 3. *Let* $\Sigma = (72)$, $\Delta = (71)$, $\Upsilon = (74)$, *and* $\Gamma = (73)$. *Then we have*

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

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

$$\neg Initiates(e, f, s_1, s_2) \tag{75}$$

$$Terminates(e, f, s_1, s_2) \leftrightarrow (e = EatFood \wedge f = FoodPresent) \tag{76}$$

$$\neg Releases(e, f, s_1, s_2) \tag{77}$$

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

$$Happens(e, s_1, s_2) \leftrightarrow (e = EatFood \wedge S(s_1, s_2) \wedge HoldsAt(FoodPresent, s_1)) \tag{78}$$

From (73), (74), and (78), we have $Happens(EatFood, S_0, S_1)$. From this, $Terminates(EatFood, FoodPresent, S_0, S_1)$ (which follows from (76)), and BDEC9, we have $\neg HoldsAt(FoodPresent, S_1)$. □

7 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).[4] We add the predicate $Actual(s)$, which represents that s is an actual situation, and the predicate $ActuallyHappens(e, s)$, which represents that event e actually occurs in situation s. We add several axioms. If a situation is actual, then a predecessor of that situation is actual:

$$Actual(s_2) \wedge S(s_1, s_2) \rightarrow Actual(s_1) \tag{79}$$

[4]See also the proposal of Baral, Gelfond, and Provetti (1997) in which action language $\mathcal{A}$ (Gelfond & Lifschitz, 1993) is modified to make an explicit distinction between hypothetical and actual actions.

A situation has at most one actual successor:

$$S(s, s_1) \land S(s, s_2) \land \mathit{Actual}(s_1) \land \mathit{Actual}(s_2) \rightarrow s_1 = s_2 \tag{80}$$

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

$$\mathit{ActuallyHappens}(e, s_1) \leftrightarrow \exists s_2\, (\mathit{Happens}(e, s_1, s_2) \land \mathit{Actual}(s_2)) \tag{81}$$

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

$$\mathit{Actual}(S_1) \tag{82}$$

From this, $S(S_0, S_1)$ (which follows from (53) and BDEC12), and (79), we have $\mathit{Actual}(S_0)$. From $S(S_0, S_1)$, $S(S_0, S_2)$ (which follows from (53) and BDEC12), (82), and (80), we have $\neg\mathit{Actual}(S_2)$. From this, (53), and (81), we have $\neg\mathit{ActuallyHappens}(\mathit{Wait}, S_0)$. From $\mathit{Happens}(\mathit{Shoot}, S_0, S_1)$ (which follows from (53)), (82), and (81), we have $\mathit{ActuallyHappens}(\mathit{Shoot}, S_0)$.

8 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 $T_1 < T_2 \lor T_1 = T_2 \lor T_2 < T_1$. They then replace this axiom with the branching time axiom $(T_1 < T_3 \land T_2 < T_3) \rightarrow (T_1 < T_2 \lor T_1 = T_2 \lor T_2 < T_1)$. 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*. In Section 5, we specify the precise relationship between BDEC and the situation calculus. We show that, if BDEC is appropriately restricted, it is equivalent to the situation calculus.

BDEC is different from the formulation of the situation calculus of Reiter and colleagues (Lin & Reiter, 1994; Pinto, 1994, 1998; Pirri & Reiter, 1999; Reiter, 1991, 1993, 2001). Pirri and Reiter (1999) provide the following foundational axioms for the situation calculus:

$$do(a_1, s_1) = do(a_2, s_2) \supset a_1 = a_2 \wedge s_1 = s_2 \tag{83}$$

$$(\forall P).P(S_0) \wedge (\forall a, s)[P(s) \supset P(do(a, s))] \supset (\forall s)P(s) \tag{84}$$

$$\neg(s \sqsubset S_0) \tag{85}$$

$$s \sqsubset do(a, s') \equiv s = s' \vee s \sqsubset s' \tag{86}$$

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 $Happens(E_1, S_0, S_1) \wedge Happens(E_2, S_0, S_1) \wedge E_1 \neq E_2$. The analogous situation calculus formula $S_1 = do(E_1, S_0) \wedge S_1 = do(E_2, S_0) \wedge E_1 \neq E_2$ is inconsistent with axiom (83). Multiple events between situations are also ruled out by Davis's (1994) axiom $result(S1, EA, S2) \wedge result(S1, EB, S2) \rightarrow EA = EB$, and by axiom BDECS11 of our restricted version of BDEC.

Based on previous proposals (Schubert, 1990; Gelfond et al., 1991; Pinto, 1994; Lin & Shoham, 1995), 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 $S_1 = do(\{A_1, A_2\}, S_0)$. 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 $Occurs(e, s)$ represents that event e occurs in situation s, the function $Next(s)$ represents the next situation after s, and the function $Result(e, s)$ represents the situation that results from e occurring in situation s. These are related via the axiom

$$Occurs(e, s) \rightarrow Next(s) = Result(e, s) \tag{87}$$

We can represent that two events occur in S_0: $Occurs(E_1, S_0) \wedge Occurs(E_2, S_0) \wedge E_1 \neq E_2$. From this and (87), we have $Next(S_0) = Result(E_1, S_0)$ and $Next(S_0) = Result(E_2, S_0)$. Thus as in BDEC, we can have distinct events E_1 and E_2 between two situations:

$$S_1 = Result(E_1, S_0) = Result(E_2, S_0) \tag{88}$$

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

$$S_2 = \mathit{Result}(E_1, S_0) \tag{89}$$
$$S_2 = \mathit{Result}(E_3, S_0) \tag{90}$$
$$S_1 \neq S_2 \tag{91}$$

The formulas (89) and (91) are inconsistent with (88). 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 (87) 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.

9 Conclusions

We introduced BDEC, a branching time discrete version of the classical logic event calculus. We proved that a restricted version of BDEC is equivalent to the situation 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., & Provetti, A. (1997). Representing actions: Laws, observations and hypotheses. *Journal of Logic Programming*, *31*(1–3), 201–243.

Clark, K. L., & Tärnlund, S.-A. (1977). A first order theory of data and programs. In B. Gilchrist (Ed.), *Proceedings of the IFIP Congress 77* (pp. 939–944). Amsterdam: North-Holland.

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

Doherty, P., Gustafsson, J., Karlsson, L., & Kvarnström, J. (1998). TAL: Temporal Action Logics language specification and tutorial. *Linköping Electronic Articles in Computer and Information Science*, *3*(015).

Gelfond, M., & Lifschitz, V. (1993). Representing action and change by logic programs. *Journal of Logic Programming*, *17*(2–4), 301–321.

Gelfond, M., Lifschitz, V., & Rabinov, A. (1991). What are the limitations of the situation calculus? In R. S. Boyer (Ed.), *Automated reasoning: Essays in honor of Woody Bledsoe* (pp. 167–179). Dordrecht: Kluwer.

Guha, R. V. (1992). *Contexts: A formalization and some applications.* Unpublished doctoral dissertation, Computer Science Department, Stanford University, Stanford, CA.

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

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

Lévy, F., & Quantz, J. J. (1998). Representing beliefs in a situated event calculus. In H. Prade (Ed.), *Proceedings of the Thirteenth European Conference on Artificial Intelligence* (pp. 547–551). Chichester, UK: John Wiley.

Lifschitz, V. (1987). Formal theories of action. In F. M. Brown (Ed.), *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop* (pp. 35–57). Los Altos, CA: Morgan Kaufmann.

Lifschitz, V. (1994). Circumscription. In D. M. Gabbay, C. J. Hogger, & J. A. Robinson (Eds.), *Handbook of logic in artificial intelligence and logic programming* (Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning, pp. 298–352). Oxford: Oxford University Press.

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

Lin, F., & Shoham, Y. (1995). Provably correct theories of action. *Journal of the ACM*, *42*(2), 293–320.

Manzano, M. (1996). *Extensions of first order logic.* Cambridge: Cambridge University Press.

McCarthy, J. (1963). *Situations, actions and causal laws.* Memo 2, Stanford, CA: Stanford Artificial Intelligence Project, Stanford University. (Memo 2, Stanford, CA: Stanford Artificial Intelligence Project, Stanford University)

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 R. Bajcsy (Ed.), *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 555–562). San Mateo, CA: Morgan Kaufmann.

McCarthy, J. (1997). *Situation calculus with concurrent events and narrative.* Stanford, CA: Department of Computer Science, Stanford University, `http://www-formal.stanford.edu/jmc/narrative/narrative.html`. (`http://www-formal.stanford.edu/jmc/narrative/narrative.html`)

McCarthy, J. (2002). Actions and other events in situation calculus. In D. Fensel, F. Giunchiglia, D. McGuinness, & M.-A. Williams (Eds.), *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning* (pp. 615–626). San Francisco: Morgan Kaufmann.

McCarthy, J., & Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer & D. Michie (Eds.), *Machine intelligence 4* (pp. 463–502). Edinburgh, Scotland: Edinburgh University Press.

Miller, R., & Shanahan, M. (2002). Some alternative formulations of the event calculus. In

A. C. Kakas & F. Sadri (Eds.), *Computational logic: Logic programming and beyond: Essays in honour of Robert A. Kowalski, part II* (Vol. 2408, pp. 452–490). Berlin: Springer.

Morgenstern, L. (2001). Mid-sized axiomatizations of commonsense problems: A case study in egg cracking. *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 V. Barr & Z. Markov (Eds.), *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference* (pp. 147–152). Menlo Park, CA: AAAI Press.

Mueller, E. T. (2006a). *Commonsense reasoning*. San Francisco: Morgan Kaufmann.

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

Nossum, R., & Thielscher, M. (1999). Counterfactual reasoning by means of a calculus of narrative context. In P. Bouquet, L. Serafini, P. Brézillon, M. Benerecetti, & F. Castellani (Eds.), *Modeling and using context* (Vol. 1688, pp. 495–498). Berlin: Springer.

Pinto, J. A. (1994). *Temporal reasoning in the situation calculus*. Unpublished doctoral dissertation, Graduate Department of Computer Science, University of Toronto, Toronto.

Pinto, J. A. (1998). Occurrences and narratives as constraints in the branching structure of the situation calculus. *Journal of Logic and Computation*, *8*, 777–808.

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

Pirri, F., & 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: From situation calculus to event calculus. *Computational Intelligence*, *12*, 478–498.

Reiter, R. (1991). The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz (Ed.), *Artificial intelligence and mathematical theory of computation: Papers in honor of John McCarthy* (pp. 359–380). San Diego: Academic Press.

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 L. C. Aiello, J. Doyle, & S. C. Shapiro (Eds.), *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning* (pp. 2–13). San Francisco: Morgan Kaufmann.

Reiter, R. (2001). *Knowledge in action: Logical foundations for specifying and implementing dynamical systems*. Cambridge, MA: MIT Press.

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

Schubert, L. K. (1990). Monotonic solution of the frame problem in the situation calculus: An efficient method for worlds with fully specified actions. In H. E. Kyburg, Jr., R. P. Loui, & G. N. Carlson (Eds.), *Knowledge representation and defeasible reasoning* (pp. 23–67). Dordrecht: Kluwer.

Shanahan, M. (1996). Robotics and the common sense informatic situation. In W. Wahlster (Ed.), *Proceedings of the Twelfth European Conference on Artificial Intelligence* (pp. 684–688). Chichester, UK: John Wiley.

Shanahan, M. (1997). *Solving the frame problem.* Cambridge, MA: MIT Press.

Shanahan, M. (1999a). The event calculus explained. In M. J. Wooldridge & M. M. Veloso (Eds.), *Artificial intelligence today: Recent trends and developments* (Vol. 1600, pp. 409–430). Berlin: Springer.

Shanahan, M. (1999b). The ramification problem in the event calculus. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (pp. 140–146). San Mateo, CA: Morgan Kaufmann.

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., & De Schreye, D. (1997). On the relation between situation calculus and event calculus. *Journal of Logic Programming*, *31*(1–3), 3–37.