Hybrid Temporal Situation Calculus *

Vitaliy Batusov[1], Giuseppe De Giacomo[2], and Mikhail Soutchanski[3]

[1] York University, Toronto, Canada
vbatusov@cse.yorku.ca
[2] Sapienza Università di Roma, Rome, Italy
degiacomo@dis.uniroma1.it
[3] Ryerson University, Toronto, Canada
mes@scs.ryerson.ca

Abstract. We present a hybrid discrete-continuous extension of Reiter's temporal situation calculus, directly inspired by hybrid systems in control theory. While keeping to the foundations of Reiter's approach, we extend it by adding a time argument to all fluents that represent continuous change. Thereby, we ensure that change can happen not only because of actions, but also due to the passage of time. We present a systematic methodology to derive, from simple premises, a new group of axioms which specify how continuous fluents change over time within a situation. We study regression for our new hybrid action theories and demonstrate what reasoning problems can be solved. Finally, we show that our hybrid theories indeed capture hybrid automata.

Keywords: Situation calculus · Temporal reasoning · Hybrid systems.

1 Introduction

Adding time and continuous change to situation calculus (SC) action theories has attracted a lot of interest over the years. A seminal book [16], refining the ideas of [13], extends situation calculus with continuous time. For each continuous process, there is an action that initiates the process at a moment of time, and there is an action that terminates it. A basic tenet of Reiter's temporal SC is that all changes in the world, including continuous processes such as a vehicle driving in a city or water flowing down a pipe, are the result of named discrete actions. Consequently, in his temporal extension of SC, fluents remain atemporal, while each instantaneous action acquires a time argument. As a side effect of this ontological commitment, continuously varying quantities do not attain values until the occurrence of a time-stamped action. For example, in Newtonian physics, suppose a player kicks a football, sending it on a ballistic trajectory. The question might be: given the vector of initial velocity, when will the ball be within 10% of the peak of its trajectory? In order to answer such questions either a natural, or an exogenous action, depending on the query, has to occur to deem the moment of interest for the query. Thus, before one can answer such questions, one needs the ability to formulate queries about the height of the ball

* Supported by the Natural Sciences and Engineering Research Council of Canada.

at arbitrary time-points, which is not directly possible without an explicit action with a time argument, if the query is formed over atemporal fluents.

In Reiter's temporal SC, to query about the values of physical quantities in between the actions (agent's or natural), one could opt for an auxiliary exogenous action $watch(t)$ [18], whose purpose is to fix a time-point t to a situation when it occurs, and then pose an atemporal query in the situation which results from executing $watch(t)$. Similarly, one can introduce an exogenous action $waitFor(\phi)$ that is executed at a moment of time when the condition ϕ becomes true, where ϕ is composed of functional fluents that are interpreted as continuous functions of time. This approach has proved to be quite successful in cognitive robotics [8] and was used to provide a SC semantics for continuous time variants of the popular planning language PDDL [3].

In this paper we study a new variant of temporal SC in which we can *directly* query continuously changing quantities at arbitrary points in time without introducing any actions (natural or exogenous or auxiliary) that supply the moment of time. Our approach is query-independent. For doing so we take inspiration from the work on hybrid systems in control theory [4, 12], which are based on discrete transitions between states that continuously evolve over time. Following this idea, the crux of our proposal is to add a new kind of axioms called *state evolution axioms* (SEA) to Reiter's successor state axioms (SSA). The SSA specify, as usual, how fluents change when actions are executed. Informally, they characterize transitions between different states due to actions. The state evolution axioms specify how the flow of time can bring changes in system parameters within a given situation while no actions are executed. Thus, we maintain the fundamental assumption of SC that all *discrete* change is due to actions, though situations now include a temporal evolution.

Reiter [16] shows how the SSA can be derived from the effect axioms in normal form by making the causal completeness assumption. We do similar work w.r.t. state evolution axioms, thus providing a precise methodology for axiomatization of continuous processes in SC in the spirit of hybrid systems. One of the key results of SC is the ability to reduce reasoning about a future situation to reasoning about the initial state by means of regression [16]. We show that a suitable notion of regression can be defined despite the continuous evolution within situations.

In hybrid automata, while continuous change is dealt with thoroughly, the discrete description is limited to finite state machines, i.e., it is based on a propositional representation of the state. SC, instead, is based on a relational representation. There are practical examples that call for an extension of hybrid systems where states have an internal relational structure and the continuous flow of time determines the evolution within the state [20]. Our proposal can readily capture these cases, by providing a relational extension to hybrid automata, which benefits from the representational richness of SC. Thus, our work can help to bring together KR and Hybrid Control, getting from the former the semantic richness of relational states and from the latter a convenient treatment of continuous time. The proofs of our thereoms are available in [1].

2 Background

Situation Calculus The situation calculus has three basic sorts (situation, action, object); formulas can be constructed over terms of these sorts. Reiter [16] shows that to solve many reasoning problems about actions, it is convenient to work with SC *basic action theories* (BATs) that have the following ingredients. For each action function $A(\bar{x})$, an *action precondition axiom* (APA) has the syntactic form $Poss(A(\bar{x}), s) \leftrightarrow \Pi_A(\bar{x}, s)$, meaning that the action $A(\bar{x})$ is possible in situation s if and only if $\Pi_A(\bar{x}, s)$ holds in s, where $\Pi_A(\bar{x}, s)$ is a formula with free variables among $\bar{x} = (x_1, \ldots, x_n)$ and s. A situation is a first-order (FO) term describing a unique sequence of actions. The constant S_0 denotes the *initial situation*, the function $do(\alpha, \sigma)$ denotes the situation that results from performing action α in situation σ, and $do([\alpha_1, \ldots, \alpha_n], S_0)$ denotes the situation obtained by consecutively performing $\alpha_1, \ldots, \alpha_n$ in S_0. The notation $\sigma' \sqsubseteq \sigma$ means that either situation σ' is a subsequence of σ or $\sigma = \sigma'$. The abbreviation *executable*(σ) captures situations σ all of whose actions are consecutively possible. Objects are FO terms other than actions and situations that depend on the domain of application. Above, $\Pi_A(\bar{x}, s)$ is a formula *uniform* in situation argument s: it talks only about situation s and uses only domain-specific predicates (see [16]). For each relational fluent $F(\bar{x}, s)$ and each functional fluent $f(\bar{x}, s)$, respectively, a *successor state axiom* (SSA) has the form

$$F(\bar{x}, do(a, s)) \leftrightarrow \Phi_F(\bar{x}, a, s) \quad \text{or} \quad f(\bar{x}, do(a, s)) = y \leftrightarrow \phi_f(\bar{x}, y, a, s),$$

where $\Phi_F(\bar{x}, a, s)$ and $\phi_f(\bar{x}, y, a, s)$ are formulas uniform in s. A BAT $\mathcal{D}$ also contains the *initial theory*: a finite set $\mathcal{D}_{S_0}$ of FO formulas uniform in S_0. Finally, BATs include a set $\mathcal{D}_{una}$ of unique name axioms for actions (UNA). If a BAT has functional fluents, it is required to satisfy an explicit consistency property whereby each functional fluent is always interpreted as a function.

BATs enjoy the *relative satisfiability* property: a BAT $\mathcal{D}$ is satisfiable whenever $\mathcal{D}_{una} \cup \mathcal{D}_{S_0}$ is. This property allows one to disregard the problematic parts of a BAT, like the second order (SO) foundational axioms Σ for situations, when checking satisfiability. BATs benefit from *regression*, a reasoning mechanism for answering queries about the future (thereby solving the *projection* problem). The regression operator $\mathcal{R}$ is defined for sufficiently specific (*regressable*) queries about the future. $\mathcal{R}[\varphi]$ is obtained from a formula φ by a syntactic manipulation (see Defn. 4.7.4 in [16]). By a seminal result in [16], regression reduces SO entailment from a BAT $\mathcal{D}$ to FO entailment by compiling dynamic aspects of the theory into the query.

To accommodate time, Reiter adds a temporal argument to all actions and introduces two special functions: *time*(a) refers to the time of occurrence of the action a, and *start*(s) refers to the starting time of situation s, i.e., the time of the latest action of s. The points constituting the timeline with dense linear order are assumed to have the standard interpretation (along with $+$, $<$, etc [16]). To model exogenous events, Reiter develops a theory of *natural actions* — non-agent actions that occur spontaneously as soon as their precondition is satisfied. Such actions are marked using the symbol $natural$, and their semantics are encoded by a modification of $executable(s)$. We use natural actions to induce relational change based on the values of the continuous quantities.

Hybrid Systems Hybrid automata are mathematical models used ubiquitously in control theory for analyzing dynamic systems which exhibit both discrete and continuous dynamics. [4] define a *basic hybrid automaton* (HA) as a system H consisting of: a finite set Q of *discrete states*; a *transition relation* $E \subseteq Q \times Q$; a *continuous state space* $X \subseteq \mathbb{R}^n$; for each $q \in Q$, a *flow function* $\varphi_q : X \times \mathbb{R} \mapsto X$ and a set $Inv_q \subseteq X$ called the *domain of permitted evolution*; for each $(q, q') \in E$, a *reset relation* $R_{q,q'} \subseteq X \times \mathcal{P}(X)$; a set $Init \subseteq \cup_{q \in Q}(\{q\} \times Inv_q)$ of *initial states*.

Like a discrete automaton, a HA has discrete states and a state transition graph, but within each discrete state its continuous state evolves according to a particular flow, e.g., it can be an implicit solution to a system of differential equations. The domain of permitted evolution delineates the boundaries which the continuous state X of the automaton cannot cross while in state q, i.e., $\varphi_q(X, t) \in Inv_q$. The reset relation helps to model discrete jumps in the value of the continuous state which accompany discrete state switching. A *trajectory* of a hybrid automaton H is a sequence $\eta = \langle \Delta_i, q_i, \nu_i \rangle_{i \in I}$, with $I = \{1, 2, \ldots\}$, where Δ_i is the *duration*, q_i is a state from Q, and $\nu_i : [0, \Delta_i] \mapsto X$ is a continuous curve along the flow φ_{q_i} (refer to [4] for details). A trajectory captures an instance of a legal evolution of a hybrid automaton over time. Duration Δ_i is the time spent by the automaton in the i-th discrete state it reaches while legally traversing the transition graph, obeying the reset relation. A trajectory is *finite* if it contains a finite number $|I|$ of steps and the sum $\Sigma_{i \in I} \Delta_i$ is finite.

3 Hybrid Temporal Situation Calculus

In our quest for a hybrid temporal SC, we reuse the temporal machinery introduced into BATs by Reiter, namely: all actions have a temporal argument and the functions $time$ and $start$ are axiomatized as before. We preserve atemporal fluents, but no longer use them to model continuously varying physical quantities. Rather, atemporal fluents serve to specify the context in which continuous processes operate. For example, the fluent $Falling(b, s)$ holds if a ball b is in the process of falling in situation s, indicating that, for the duration of s, the position of the ball should be changing as a function of time according to the equations of free fall. The fluent $Falling(b, s)$ may be directly affected by instantaneous actions $drop(b, t)$ (ball begins to fall at the moment t) and $catch(b, t)$ (ball stops at t), but the effect of these actions on the position of the ball comes about only indirectly, by changing the context of a continuous trajectory and thus switching the continuous trajectory that the ball can follow. Thus, a falling ball is one context, and a ball at rest is another. In general, there are finitely many parametrized context types which are pairwise mutually exclusive when their parameters are appropriately fixed, and each context type is characterized by its own continuous function that determines how a physical quantity changes.

To model continuously varying physical quantities, we introduce new functional fluents with a temporal argument. We imagine that these fluents can change with time, and not only as a direct effect of actions. For example, for the context where the ball is falling, the velocity of the ball at time t represented by fluent $vel(b,t,s)$ can be specified as $[Falling(b,s) \wedge y = vel(b, start(s), s) - g \cdot (t - start(s))] \rightarrow vel(b,t,s) = y$. Notice that this effect axiom does not mention actions and describes the evolution of vel within a single situation.

Deriving State Evolution Axioms Our starting point is a *temporal change axiom* (TCA) which describes a single law governing the evolution of a particular temporal fluent due to the passage of time in a particular context of an arbitrary situation. An example of a TCA was given above for $vel(b,t,s)$. We assume that a TCA for a temporal functional fluent f has the general syntactic form

$$\gamma(\bar{x}, s) \wedge \delta(\bar{x}, y, t, s) \rightarrow f(\bar{x}, t, s) = y, \tag{1}$$

where t, s, $\bar{x}$, y are variables and $\gamma(\bar{x}, s)$, $\delta(\bar{x}, y, t, s)$ are formulas uniform in s whose free variables are among those explicitly shown. We call $\gamma(\bar{x}, s)$ the *context*, as it specifies the condition under which the formula $\delta(\bar{x}, y, t, s)$ is to be used to compute the value of fluent f at time t. Note that contexts are time-independent. The formula $\delta(\bar{x}, y, t, s)$ encodes a function (e.g., a solution to the initial value problem for a system of the ordinary differential equations [19]) which specifies y in terms of the values of other fluents at s, t. For each TCA (1) to be *well-defined*, we require that the background theory entails $\gamma(\bar{x}, s) \rightarrow \exists y\, \delta(\bar{x}, y, t, s)$. In other words, whatever the circumstance, the TCA must supply a value for the quantity modelled by f if its context is satisfied. A set of k well-defined temporal change axioms for some fluent f can be equivalently expressed as an axiom of the form (2) below, where $\Phi(\bar{x}, y, t, s)$ is $\bigvee_{1 \le i \le k} (\gamma_i(\bar{x}, s) \wedge \delta_i(\bar{x}, y, t, s))$. For each such axiom, we require that the background theory entails the condition (3).

$$\Phi(\bar{x}, y, t, s) \rightarrow f(\bar{x}, t, s) = y, \tag{2}$$

$$\Phi(\bar{x}, y, t, s) \wedge \Phi(\bar{x}, y', t, s) \rightarrow y = y'. \tag{3}$$

Condition (3) guarantees the consistency of the axiom (2) by preventing a continuous quantity from having more than one value at any moment of time. Provided (3), we can assume w.l.o.g. that all contexts in the given set of TCA are pairwise mutually exclusive w.r.t. the background theory $\mathcal{D}$.

Having combined all laws which govern the evolution of f with time into a single axiom (2), we can make a causal completeness assumption: *there are no other conditions under which the value of f can change in s from its initial value at $start(s)$ as a function of t.* We capture this assumption formally by the explanation closure axiom

$$f(\bar{x}, t, s) \neq f(\bar{x}, start(s), s) \rightarrow \exists y\, \Phi(\bar{x}, y, t, s). \tag{4}$$

Theorem 1. *Let $\Psi(\bar{x}, s)$ denote $\bigvee_{1 \le i \le k} \gamma_i(\bar{x}, s)$. The conjunction of axioms (2) and (4) in the models of (3) is logically equivalent to*

$$f(\bar{x}, t, s) = y \leftrightarrow [\Phi(\bar{x}, y, t, s) \vee y = f(\bar{x}, start(s), s) \wedge \neg\Psi(\bar{x}, s)]. \tag{5}$$

We call the formula (5) a *state evolution axiom* (SEA) for the fluent f. Note what the SEA says: f evolves with time during s according to some law whose context is realized in s or stays constant if no context is realized. The assumption (4) simply states that all reasons for change have been already accounted for in (2) and nothing is missed. It is important to realize that $\mathcal{D}_{se}$, a set of SEAs, complements the SSAs derived in [16] using a similar technique.

Hybrid Basic Action Theories The SEA for a temporal fluent f does not completely specify the behaviour of f because it talks only about change within a single situation s. To complete the picture, we need a SSA describing how the value of f changes (or does not change) when an action is performed. A straightforward way to accomplish this would be by an axiom which would enforce continuity, e.g., $f(\bar{x}, time(a), do(a, s)) = f(\bar{x}, time(a), s)$. However, this choice would preclude the ability to model action-induced discontinuous jumps in the value of the continuously varying quantities, such as the sudden change of acceleration from 0 to $-9.8m/s^2$ when an object is dropped. To circumvent this, for each temporal functional fluent $f(\bar{x}, t, s)$, we introduce an auxiliary atemporal functional fluent $f_{init}(\bar{x}, s)$ whose value in s represents the value of the quantity modelled by f in s at the time instant $start(s)$. We axiomatize f_{init} using a SSA derived from an effect axiom for $f_{init}(\bar{x}, s)$ and a frame axiom of the form $\neg\exists y(e(\bar{x}, y, a, s)) \rightarrow f_{init}(\bar{x}, do(a, s)) = f(\bar{x}, time(a), s)$ stating that if no relevant effect is invoked by the action a, f_{init} assumes the most recent value of f. The SSA for f_{init} has standard syntax and describes how the initial value of f in $do(a, s)$ relates to its value at the same time instant in s (i.e., prior to a). To establish a consistent relationship between temporal fluents and their *init*-counterparts, we require that, in an arbitrary situation, the continuous evolution of each temporal fluent f starts with the value computed for f_{init} by its successor state axiom.

A *hybrid basic action theory* is a collection of axioms $\mathcal{D} = \Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{se}$ such that

1. Every action mentioned in $\mathcal{D}$ is temporal;
2. $\Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0}$ constitutes a BAT as per Definition 4.4.5 in [16];
3. $\mathcal{D}_{se}$ is a set of SEA of the form $f(\bar{x}, t, s) = y \leftrightarrow \psi_f(\bar{x}, t, y, s)$ where $\psi_f(\bar{x}, t, y, s)$ is uniform in s, such that $\mathcal{D}_{ss}$ contains an SSA for f_{init};
4. For each SEA of the form above, $\mathcal{D}_{una} \cup \mathcal{D}_{S_0}$ entails

$$\forall\bar{x}\forall t.\ \exists y(\psi_f(\bar{x}, t, y, s)) \wedge \forall y\forall y'(\psi_f(\bar{x}, t, y, s) \wedge \psi_f(\bar{x}, t, y', s) \rightarrow y = y'), \quad (6)$$

$$\exists y(f_{init}(\bar{x}, s) = y \wedge \psi_f(\bar{x}, start(s), y, s)); \quad (7)$$

A set $\mathcal{D}_{se}$ of SEA is *stratified* iff there are no temporal fluents $f_1, \ldots, f_n$ such that $f_1 \succ f_2 \succ \ldots \succ f_n \succ f_1$ where $f \succ f'$ holds iff there is a SEA in $\mathcal{D}_{se}$ where f appears on the left-hand side and f' on the right-hand side. A hybrid BAT is *stratified* iff its $\mathcal{D}_{se}$ is.

Theorem 2. *A stratified hybrid BAT $\mathcal{D}$ is satisfiable iff $\mathcal{D}_{una} \cup \mathcal{D}_{S_0}$ is.*

Example 1. (See [1] for an illustartion and additional details.) Consider a macroscopic urban traffic domain along the lines of [20]. For simplicity, we consider a single intersection of two 2-lane roads. Facing the intersection i are 4 incoming and 4 outgoing road segments. Depending on the traffic light, a car may turn left, turn right, or drive straight from an incoming lane to an outgoing lane. Each lane is denoted by a constant and each path through the intersection i is encoded using the predicates $st(i, r_1, r_2)$ (straight connection from lane r_1 to r_2 at intersection i), $lt(i, r_1, r_2)$ (left turn), and $rt(i, r_1, r_2)$ (right turn). The number of cars per unit of time that can pass through each connection is specified by the function $flow(i, r_1, r_2)$.

The outgoing lanes are of infinite capacity and are not modelled. The traffic lights are controlled by a simple looping automaton with the states $Green(i, r, s)$ (from lane r, go straight or turn right), followed by $RArr(i, r, s)$ (*right arrow*, i.e., only turn right), followed by $Red(i, r, s)$ (stop), and then $LArr(i, r, s)$ (only turn left), such that mutually orthogonal directions are in antiphase to each other. The switching between states for all r is triggered by the action $switch(i, t)$ with precondition $Poss(switch(i, t),\ s) \leftrightarrow start(s) \leq t$ via a set of simple SSA.

The continuous quantity we wish to model is the number of cars at intersection i queued up in lane r. For that, we use the temporal fluent $que(i, r,\ t, s)$ and its atemporal counterpart $que_{init}(i, r, s)$. Since the lane r may run dry, we call on the natural action $empty(i, r, t)$ to change the relational state:

$$
\begin{aligned}
&Poss(empty(i, r, t), s) \leftrightarrow start(s) \leq t \wedge que(i, r, t, s) = 0,\\
&a = empty(i, r, t) \wedge y = 0 \rightarrow que_{init}(i, r, do(a, s)) = y,\\
&a \neq empty(i, r, t) \wedge y = que(i, r, time(a), s) \rightarrow que_{init}(i, r, do(a, s)) = y.
\end{aligned}
$$

We can now formulate the TCA for que according to traffic rules. Cars do not move at a red light: $[Red(i, r, s) \wedge y = que_{init}(i, r, s)] \rightarrow que(i, r, t, s) = y$. When a non-empty lane r sees the left (or right) arrow, its queue decreases linearly with the rate associated with the left (resp., right) turn. For the signal $Green(i, r, s)$, the queue decreases with a combined rate of the straight connection and the right turn, i.e. $y = (que_{init}(i, r, s) - (flow(i, r, r') + flow(i, r, r'')) \cdot (t - start(s))$.

From these TCA, by Theorem 1, we obtain a SEA below (simplified for brevity). Notice that the last line comes not from the TCA but from the explanation closure (4) enforced by Theorem 1 and asserts the constancy of que in the context which the TCA did not cover (movement is allowed but the lane is empty). In general, the modeller only needs to supply the TCA for the contexts where the quantity changes with time.

$$
\begin{aligned}
&que(i, r, t, s) = y \leftrightarrow (\exists \tau \exists q_0 \exists r_L \exists r_S \exists r_R).\\
&\quad \tau = (t - start(s)) \wedge q_0 = que_{init}(i, r, s) \wedge lt(i, r, r_L) \wedge st(i, r, r_S) \wedge rt(i, r, r_R) \wedge\\
&\quad \big[LArr(i, r, s) \wedge q_0 \neq 0 \wedge y = (q_0 - flow(i, r, r_L) \cdot \tau) \vee\\
&\quad\ Green(i, r, s) \wedge q_0 \neq 0 \wedge y = (q_0 - (flow(i, r, r_S) + flow(i, r, r_R)) \cdot \tau) \vee\\
&\quad\ RArr(i, r, s) \wedge q_0 \neq 0 \wedge y = (q_0 - flow(i, r, r_R) \cdot \tau) \vee\\
&\quad\ Red(i, r, s) \wedge y = q_0 \vee \neg Red(i, r, s) \wedge q_0 = 0 \wedge y = 0\big].
\end{aligned}
$$

4 Regression

Projection is a ubiquitous computational problem concerned with establishing the truth value of a statement after executing a given sequence of actions. We solve it with the help of regression. The notions of uniform and regressable formulas trivially extend to hybrid BATs. The regression operator $\mathcal{R}$ as defined for atemporal BATs in Definition 4.7.4 of [16] can be extended to hybrid BATs in a straightforward way. When $\mathcal{R}$ is applied to a regressable formula W, $\mathcal{R}[W]$ is determined relative to a hybrid BAT. We extend $\mathcal{R}$ as follows.

Let $\mathcal{D}$ be a hybrid BAT, and let W be a regressable formula. If W is a non-fluent atom that mentions $start(do(\alpha,\sigma))$, then $\mathcal{R}[W] = \mathcal{R}[W|^{start(do(\alpha,\sigma))}_{time(\alpha)}]$. If W is a non-$Poss$ atom and mentions a functional fluent uniform in σ, then this term is either atemporal or temporal. The former case is covered by Reiter. In the latter case, the term is of the form $f(\bar{C},\tau^\star,\sigma)$ and has a SEA $f(\bar{x},t,s) = y \leftrightarrow \psi_f(\bar{x},t,y,s)$, so we rename all quantified variables in $\psi_f(\bar{x},t,y,s)$ to avoid conflicts with the free variables of $f(\bar{C},\tau^\star,\sigma)$ and define $\mathcal{R}[W]$ to be $\mathcal{R}[\exists y.\,(\tau^\star = start(\sigma) \wedge y = f_{init}(\bar{x},\sigma) \vee \tau^\star \neq start(\sigma) \wedge \psi_f(\bar{C},\tau^\star,y,\sigma)) \wedge W|^{f(\bar{C},\tau^\star,\sigma)}_y]$, where y is a new variable not occurring free in W, $\bar{C}$, $\tau^\star$, σ. Intuitively, this transformation replaces the temporal fluent f with either the value of f_{init} if f is evaluated at the time of the last action or, otherwise, with the value determined by the right-hand side of the SEA for f.

Theorem 3. *If W is a regressable sentence of SC and $\mathcal{D}$ is a stratified hybrid basic action theory, then $\mathcal{D} \models W$ iff $\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]$.*

Example 2. Let the initial state in the previous example entail the following:

$$start(S_0) = 0,\ Red(I, in_1, S_0),\ que_{init}(I, in_1, S_0) = 100,$$
$$flow(I, in_1, out_2) = 5,\ flow(I, in_1, out_3) = 15,\ flow(I, in_1, out_4) = 10.$$

Let W be $que(I, in_1, 3, \sigma) < 95$, i.e., there are fewer than 95 cars in lane in_1 at time 3 in situation σ, where σ is $do([switch(I,1), switch(I,2)], S_0)$. In this narrative, the lane in_1 sees the red light, which at $t=1$ switches to the left arrow, and at $t=2$ to green. To check if $\mathcal{D} \models W$, we use Theorem 3 to reduce W to an equivalent statement about S_0:

$$\mathcal{R}[que(I, in_1, 3, \sigma) < 95] = que_{init}(I, in_1, S_0) - 10(2-1) - (15+5)(3-2) < 95.$$

The resulting query can be answered by FO means by plugging 100 for the initial number of cars at in_1: $100 - 10 - 20 = 70 < 95$, so the statement is true.

Regression can also be a powerful diagnostic tool. By analyzing the results of partial regression of a temporal query, one can attribute its validity to a particular action of the narrative. Let $\mathcal{R}^{\sigma'}$ be a variant of $\mathcal{R}$ which does not regress beyond σ'. We can establish whether $\mathcal{R}^{\sigma'}[W]$ is true for each $\sigma' \sqsubseteq \sigma$ as before. In our example, the query holds during and after $switch(I,2)$ but is false before and at the instant of $switch(I,1)$. We conclude that the action $switch(I,1)$ as well as the time that has passed since $t=1$ up to the time when $\mathcal{R}^{do(switch(I,1),S_0)}[W]$ became true are responsible for the fact that W holds at σ. Note that W can be an arbitrary regressable property of the dynamic system.

5 Comparison with Previous Approaches

Considering that discrete-continuous systems have been a hot topic for decades, it is impossible to fairly compare hybrid situation calculus to a representative subset of all work in that area. Hence, we draw comparisons only to approaches from the same paradigm.

A seminal work by Sandewall [17] points out that discarding information from a theory cannot lead to better inferences. He argues that differential calculus is the perfect language for modelling continuous change and that the essential task in describing physical systems is to provide a logical foundation for discrete state transitions. Pinto [13] presents initial proposals to introduce time into the situation calculus; these works focuse on a so-called actual sequence of actions and introduced representation for occurrences of actions w.r.t. an external time-line. Ch. 6 of [13] discusses examples of continuous change and natural events following [17], but without using Sandewall's non-monotonic solution to the frame problem. It also introduces a class of objects called parameters that are used to name continuously varying properties such that each parameter behaves according to a unique function of time during a fixed situation. It is mentioned that parameters can be replaced with functional fluents of time, but this direction was not elaborated. Building on earlier work of [17, 13], [11] introduces time-independent fluents and situation-independent parameters that can be regarded as functions of time, but provides only an example, and no general methodology. [16] provides the modern axiomatization of time, concurrency, and natural actions in SC. However, [16] allows only atemporal fluents in contrast to [13]. For this reason, [18] proposes an auxiliary action $watch(t)$ (see below).

The example in Section 3 helps illustrate the differences with our approach. Consider Reiter's temporal SC [16]: since fluents are atemporal, the TCA above are replaced by effect axioms for the atemporal fluent $que(i, r, s)$, e.g.,

$$a = switch(i,t) \wedge \big[LArr(i,r,s) \wedge que(i,r,s) \neq 0 \wedge \exists r'[lt(i,r,r') \wedge \\ y = (que(i,r,s) - flow(i,r,r') \cdot (time(a) - start(s))]\big] \rightarrow que(i,r,do(a,s)) = y.$$

Note that, in effect axioms, the change in que is associated with a named action. The modeller must replicate this axiom for each action which might affect the context $LArr(i,r,s) \wedge que(i,r,s) \neq 0$, and likewise for all other contexts and TCA. In our approach, the change in context is handled separately and does not complicate the axiomatization of continuous dynamics. The right-hand side of the resulting SSA, $\gamma_{que}(i,r,y,s) \vee que(i,r,s) = y \wedge \neg\exists y' \gamma_{que}(i,r,y',s)$, can be obtained from the right-hand side of the SEA above by replacing t with $time(a)$, $que_{init}(i,r,s)$ with $que(i,r,s)$, and the last line by $que(i,r,s) = y \wedge \neg\exists y' \gamma_{que}(i,r,y',s)$. Notice that the expression $\gamma_{que}(i,r,y,s)$ occurs twice — first due to the effect axiom (in a normal form) and then again due to explanation closure — see examples in Section 3.2.6 in [16]. In our approach, only the essential atemporal part of that expression appears. Furthermore, Reiter's version of the precondition axiom for $empty(i,r,t)$ is necessarily cumbersome because it mentions $que(i,r,t,s)$, whose evolution (and thus the value at t) depends on the current relational state of s. Therefore, the modeller must include the right-hand side of the SSA in the precondition, thereby increasing the size of the axioms by roughly the size of the SSA for the continuous fluent F for every occurrence of F in a precondition axiom while not adding any new information. Moreover, since fluents are atemporal, evaluating them at arbitrary moments of time t requires an auxiliary action.

The approach due to [18] introduces the special action $watch(t)$ to advance time to the time-point t. This allows one to access continuous fluents in between the agent actions, but at a cost: replacing $que(i, r, t, s)$ by $que(i, r, do(watch(t), s))$ in the precondition axiom makes the right-hand side non-uniform in s, violates Defn. 4.4.3 in [16], and thus steps outside of the well-studied realm of BATs. A later proposal due to [8] considers fluents whose values range over functions of time, but neither the fluents nor the actions have a temporal argument. Domain actions occur at the same instant as the preceding situation, and the mechanism for advancing time is the special action $waitFor(\phi)$ which simulates the passage of time until the earliest time-point where ϕ holds. Aimed specifically at robotic control, this approach relies on a cc-Golog program to trigger the $waitFor$ action.

Finzi and Pirri [6] introduce *temporal flexible situation calculus*, a dialect aimed to provide formal semantics and a Golog implementation for constraint-based interval planning which requires dealing with multiple alternating timelines. To represent processes, they introduce fluents with a time argument. However, this time argument marks the instant of the process' creation and is not associated with a continuous evolution.

6 Modelling Hybrid Automata

Hybrid BATs introduced here are naturally suitable for capturing hybrid automata [12]. Given an arbitrary basic hybrid automaton H, c.f., Section 2, we proceed as follows. For every discrete state in the set Q, we introduce a constant q_i with $1 \leq i \leq |Q|$ and let $\mathcal{D}_{S_0}$ contain unique name axioms for all q_i. The transition relation E is encoded by a finite set of facts $E(q, q')$. Each flow φ_q is encoded by the function $flow$ such that $flow(q, x, t) = y$ iff $\varphi_q(x, t) = y$. Each set of invariant states Inv_q is encoded by the predicate $Inv(q, x)$ which holds iff $x \in Inv_q$. Each reset relation $R_{q,q'}$ is encoded by the predicate $R(q, q', x, y)$ which holds iff $y \in R_{q,q'}(x)$. The set of initial states $Init$ is encoded by the predicate $Init(q, x)$ which holds iff $(q, x) \in Init$.

Let $Q(s)$ denote the discrete and $X(t, s)$ the continuous state. Let $tr(q, q', y, t)$ be the action representing a transition from state q to q' at time t while resetting the continuous state to the value y. The automaton can be described as

$$Poss(tr(q, q', y, t), s) \leftrightarrow Q(s) = q \wedge E(q, q') \wedge R(q, q', X(t, s), y) \wedge Inv(q', y),$$
$$Q(do(a, s)) = q \leftrightarrow \exists q', y, t(a = tr(q', q, y, t)) \vee Q(s) = q \wedge \neg\exists q', y, t(a = tr(q, q', y, t)),$$
$$X_{init}(do(a, s)) = x \leftrightarrow \exists q \exists q' \exists t(a = tr(q, q', x, t)),$$
$$X(t, s) = x \leftrightarrow \bigvee_{i=1}^{k}[Q(s) = q_i \wedge x = flow(q_i, X_{init}(s), t)].$$

Theorem 4. *Let $\mathcal{D}$ be a satisfiable hybrid BAT axiomatizing a hybrid automaton H as above, let σ be an executable ground situation and let $\tau \geq start(\sigma)$. Then*

$$\mathcal{D} \models Init(Q(S_0), X_{init}(S_0)) \wedge (\forall a, s, t)\big[do(a, s) \sqsubseteq \sigma \wedge start(s) \leq t \leq time(a) \vee s = \sigma \wedge start(\sigma) \leq t \leq \tau\big] \rightarrow Inv(Q(s), X(s, t))$$

if and only if a finite trajectory of H can be uniquely constructed from σ and τ.

Clearly, this axiomatization rules out non-trivial queries about the content of the states because its discrete states are a finite set without objects, relations, etc. A general hybrid BAT does not have this limitation. While classic HA are based on a finite representation of states and atomic state transitions, richer representations began to attract the interest of the hybrid system community. Of particular interest is the work by Platzer [15] based on FO dynamic logic extended to handle differential equations for describing continuous change. Our work contributes to this line of research by providing a very rich representation of the discrete states described relationally in FOL. Both [14] and our paper propose to go beyond finite-state HA. The key advantage of our work is in the availability of situation terms, and therefore, the regression operator. Thus, the usual SC-based reasoning tasks [16] can be solved in our hybrid BATs.

7 Conclusion

Inspired by hybrid systems, we have proposed a temporal extension of SC with a clear distinction between atemporal fluents, responsible for transitions between states, and temporal fluents, representing continuous change within a state. While this paper focuses on semantics, the connection with hybrid systems established here opens new perspectives for future work on automated reasoning as well. In hybrid systems, the practical need for robust specification and verification tools for HA resulted in the development of a multitude of logic-based approaches (see [4] for an overview). More recently, [7] show that certain classes of decision problems belong to reasonable complexity classes. These results provide foundations for verification of robustness in hybrid systems [9]. Platzer's work offers some decidability results for verification based on quantifier eliminations [14, 15]. Note that the quantified differential dynamic logic [14], which focuses on functions and does not allow for arbitrary relations on objects, cannot encode SC action theories in an obvious way, i.e., it includes only one primitive action (assignment), but BATs provide agent actions that can model a system at a higher level of abstraction. Nevertheless, it may be interesting to study the reductions of fragments of Golog [10] and BATs with or without continuous time to such a dynamic logic, to exploit existing [14] and future decidability results.

On the other hand, while research in hybrid systems focuses on certain verification problems, the present paper, due to regression, proposes an approach to solve other reasoning problems that cannot be formulated in hybrid systems. Recent work on bounded theories [5, 2] provides promising means to study decidable cases in this realm, which could be of interest to hybrid systems as well.

References

1. Batusov, V., De Giacomo, G., Soutchanski, M.: Hybrid temporal situation calculus. CoRR **1807.04861** (2018), https://export.arxiv.org/abs/1807.04861
2. Calvanese, D., De Giacomo, G., Montali, M., Patrizi, F.: First-order μ-calculus over generic transition systems and applications to the situation calculus. Inf. Comput. **259**(3), 328–347 (2018)
3. Claßen, J., Hu, Y., Lakemeyer, G.: A situation-calculus semantics for an expressive fragment of PDDL. In: Proceedings of AAAI-2007, July 22-26, 2007, Vancouver, British Columbia, Canada. pp. 956–961. AAAI Press (2007)
4. Davoren, J., Nerode, A.: Logics for hybrid systems (invited paper). Proceedings of the IEEE **88**(7), 985–1010 (2000)
5. De Giacomo, G., Lespérance, Y., Patrizi, F.: Bounded situation calculus action theories. Artif. Intell. **237**, 172–203 (2016)
6. Finzi, A., Pirri, F.: Representing flexible temporal behaviors in the situation calculus. In: Proceedings of IJCAI-05, Edinburgh, UK, 2005. pp. 436–441 (2005)
7. Gao, S., Avigad, J., Clarke, E.M.: Delta-decidability over the reals. In: Lipovac, V., Scedrov, A. (eds.) Proceedings of LICS-2012, Dubrovnik, Croatia, June 25-28, 2012. pp. 305–314. IEEE Computer Society (2012)
8. Grosskreutz, H., Lakemeyer, G.: cc-Golog – A logical language dealing with continuous change. Logic Journal of the IGPL **11**(2), 179–221 (2003)
9. Kong, S., Gao, S., Chen, W., Clarke, E.M.: dReach: δ-reachability analysis for hybrid systems. In: Baier, C., Tinelli, C. (eds.) Proceedings of TACAS 2015, Held as Part of ETAPS 2015, London, UK, April 11-18, 2015. Lecture Notes in Computer Science, vol. 9035, pp. 200–205. Springer (2015)
10. Levesque, H.J., Reiter, R., Lespérance, Y., Lin, F., Scherl, R.B.: GOLOG: A logic programming language for dynamic domains. J. Log. Program. **31**, 59–83 (1997)
11. Miller, R.: A case study in reasoning about actions and continuous change. In: Wahlster, W. (ed.) Proceedings of ECAI'96. pp. 624–628 (1996)
12. Nerode, A.: Logic and control. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) Proceedings of CiE 2007, Siena, Italy, June 18-23, 2007. Lecture Notes in Computer Science, vol. 4497, pp. 585–597. Springer (2007)
13. Pinto, J.: Temporal Reasoning in the Situation Calculus. Ph.D. thesis, University of Toronto, Toronto, Canada (1994)
14. Platzer, A.: A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems. Logical Methods in Computer Science **8**(4) (2012)
15. Platzer, A.: A complete uniform substitution calculus for differential dynamic logic. J. Autom. Reasoning **59**(2), 219–265 (2017)
16. Reiter, R.: Knowledge in action: logical foundations for specifying and implementing dynamical systems. MIT press Cambridge (2001)
17. Sandewall, E.: Combining logic and differential equations for describing real-world systems. In: Brachman, R.J., Levesque, H.J., Reiter, R. (eds.) Proceedings of KR'89. Toronto, Canada, May 15-18, 1989. pp. 412–420. Morgan Kaufmann (1989)
18. Soutchanski, M.: Execution monitoring of high–level temporal programs. In: Beetz, M., Hertzberg, J. (eds.) Robot Action Planning, Proceedings of the IJCAI-99 Workshop. pp. 47–54. Stockholm, Sweden (1999)
19. Teschl, G.: Ordinary Differential Equations and Dynamical Systems. AMS (2012), https://www.mat.univie.ac.at/ gerald/ftp/book-ode/ode.pdf
20. Vallati, M., Magazzeni, D., De Schutter, B., Chrpa, L., McCluskey, T.L.: Efficient macroscopic urban traffic models for reducing congestion: A PDDL+ planning approach. In: AAAI. pp. 3188–3194 (2016)