An Illustrated Situation Calculus Abstraction for Iterative Explanatory Diagnosis

Christine Task[1], Mark Wilson[2], Matthew Molineaux[1], and **David W. Aha[2]**
[1]Knexus Research Corporation; Springfield, VA; USA
[2]Navy Center for Applied Research in AI; Naval Research Laboratory (Code 5514); Washington, DC; USA
{*first.last*}@knexusresearch.com | {*first.last*}@nrl.navy.mil

Abstract

Effective goal reasoning requires a robust estimation of the state of the agent's environment, which can be challenging to achieve in partially observable domains. Iterative explanatory diagnosis allows a goal reasoning agent to infer important information that is not directly observable, but in realistic domains this requires repeatedly searching a large, complex solution space over two interdependent sources of uncertainty: unobservable events in the agent's history and unobservable fluents in the hidden state. We propose a formal abstraction of explanatory diagnosis that provides a convenient visual representation of the relation between these sources of uncertainty. Our abstraction combines McIlraith's [1997] situation-calculus dialect for diagnosis and Pang and Holte's [2011] model of state-set graphs. For this initial work, we assume a deterministic environment: all events are determined by agent actions and the partially observable initial environmental state.

1 Introduction

Goal reasoning is the ability of an agent to deliberate about and alter its goals during online plan execution in complex domains. One common trigger for goal reasoning is a discrepancy between the observed state of the environment during execution and the expected state that was predicted during planning [Vattam *et al*. 2013]. Such discrepancies often represent an opportunity to reason about hidden information in a partially observable domain. Consider a goal reasoning robotic agent assisting a military squad; if it observes gunfire sounds and troop movements that were not expected in the predicted environmental state, then an explanation which resolves these discrepancies can allow it to infer the presence of a hidden, unobservable sniper. This inference about important, hidden state information allows the agent to select an appropriate goal and accurately plan to achieve that goal in partially observable domains. We refer to this reasoning task as the *explanatory diagnosis* problem. Explanatory diagnosis is used in goal reasoning agents such as ARTUE [Molineaux *et al*. 2010] to provide a more accurate understanding of the environment, allowing the agent to make more accurate predictions in future plans, and select appropriate goal. When the goal reasoning agent cannot explain discrepancies in its observations, it is unable to respond to unobservable phenomena in its environment.

To solve the explanatory diagnosis problem, an agent must address two sources of uncertainty introduced by partial observability in the domain: (1) uncertainty about hidden facts in the current state and (2) uncertainty about the set of agent actions and environmental events that led to this state (the agent's *execution history*). *Iterative explanatory diagnosis* is a complex iterative inference task that the agent must perform repeatedly. In each iteration the agent improves a running hypothesis about the execution history (its *explanation*) by reconciling it with partial observations, and then applies it to estimate unobservable values in the current state. The interlocking spaces of possible states and possible execution histories are inherently interdependent, creating a complex solution space for the iterative explanatory diagnosis problem. Little work exists on formally describing explanatory diagnosis in the goal reasoning context. To better understand the solution space of this problem, we develop a formal abstraction that provides a convenient visual representation of the relationship. Our formalization employs McIlraith's [1997] situation-calculus dialect from diagnosis and Pang and Holte's [2011] model of state-set graphs. For this initial work, we assume a deterministic environment: all events are determined by agent actions and the partially observable initial environmental state.

We begin in §2 by discussing the related work that our model is derived from, and then in §3 briefly introduce the three core components of our abstraction: the iterative explanatory diagnosis problem, the situation calculus, and state-set graphs. In §4 we then describe our abstraction for the iterative explanatory diagnosis problem, in which we employ the situation calculus to characterize the space of possible execution histories and apply the state-set framework to characterize the space of possible states. We conclude in §5 by discussing the implications of this abstraction for the design of iterative explanatory diagnosis algorithms.

2 Related Work

In realistic environments, reasoning agents encounter many forms of uncertainty. Explanatory diagnosis is concerned with inferring an accurate execution history for the agent: inferring the sequence of states, agent actions and resulting environmental events leading from the initial state to the current state. However, uncertainty occurs at a lower level as well (for example, in noisy sensor readings) leading to uncertain values for state variables. This form of uncertainty

is beyond the scope of this work, but cyclical projection/observation/belief revision algorithms exist to address it as well, and are well known. These include Kalman Filters [Kalman 1960], and more generally, Particle Filters [Thrun et al. 2005].

Our work, however, primarily relates to prior efforts on iterative explanation in goal reasoning, the situation calculus in explanatory diagnosis, and the state-set framework. We briefly review this literature here and provide greater depth on these topics in §3.

The ARTUE agent [Molineaux *et al.* 2010] is an implementation of the goal-driven autonomy model of goal reasoning; it attempts to explain discrepancies between the agent's expectations and observations during plan execution. ARTUE originally employed a non-iterative diagnosis procedure based on an Assumption-Based Truth Maintenance System [De Kleer 1986] that maintained only one historical observation. More recent versions, such as FOOLMETWICE [Molineaux and Aha 2014], employ the DiscoverHistory algorithm [Molineaux and Aha 2015], which reasons about a larger history of observations to eliminate incorrect hypotheses/explanations. While DiscoverHistory provides a specific search algorithm for producing consistent execution histories, we instead characterize the general explanatory diagnosis problem using formal models of state sets and the situation calculus.

The situation calculus [Lin and Reiter 1994] is a formal model of execution histories in planning agents. McIlraith [1997] proposes a dialect of the situation calculus tailored to the application of explanatory diagnosis, and we adapt this dialect to formally model explanatory diagnosis in online goal reasoning agents.

Pang and Holte [2011] introduce their state-set framework to provide a mechanism for visualizing and formally analyzing transitions (and sequences of transitions) between *sets* of states. While this concept applies to several search and abstraction problems in planning, we will employ it to model sets of possible states.

3 Background

To provide background for our abstraction, we briefly introduce the explanatory diagnosis problem, the situation calculus, and the state-set framework. We first describe the iterative explanatory diagnosis problem as it occurs in a goal reasoning agent. The situation calculus provides a formal model for sequences of actions in an execution history, and we employ a variant designed for explanatory diagnosis by McIlraith [1997]. In particular, we apply it to a goal reasoning context. Finally, we use the state-set framework to abstract sets of possible previous states, given the actions and events in the agent's execution history.

3.1 The Iterative Explanatory Diagnosis Problem

We consider the diagnostic needs of a goal reasoning agent that interacts with a partially observable environment; its objective is to select appropriate goals at different points in its execution, identify viable plans (i.e., sequences of actions) to achieve the selected goals, and successfully complete the chosen plans. We characterize the agent's interaction with a partially observable, deterministic, dynamic environment as a cycle comprised of three phases: Agent Action, Environmental Events, and Agent Observation. Over the course of plan execution, each agent action alters the environment (Agent Action). These alterations can trigger in the environment a sequence of exogenous, unobservable events that further alter it (Environmental Events). Events occur deterministically when their (potentially unobservable) preconditions are satisfied, and multiple events may co-occur (forming an *event set)*. After taking each action, and after any resulting environmental events have been resolved, the agent makes an observation (Agent Observation) which provides it with the current values of all observable facts about the current state. However, there are *hidden* facts whose values the agent cannot directly observe. In general, we use the catch-all term *fluent* to refer to facts about the environmental state (which may be represented by a predicate or function); the current state value is defined by the current values of all fluents, both hidden and observable.

Thus the agent has limited knowledge: it knows with certainty which actions it has taken itself, and it possesses models of the preconditions and effects for all actions and events. But it must work with two forms of uncertainty: it cannot directly observe the subsequent state transitions (events) its actions trigger in the environment, and it cannot observe the complete state of the environment.

In order to select a goal and plan a sequence of actions (and expected environmental events) to achieve it, the agent needs an estimate of the complete current state, including the hidden fluent values. One way of estimating this state is to maintain a hypothesis about the initial state of the system and the execution history (including expected event-set sequences). Thus, before it makes each observation, the agent possesses an *expected* value for each fluent (both observable and hidden). If the observation conflicts with the agent's estimated value for an observable fluent (an *inconsistency*), then an unexpected event sequence must have occurred[1].

This is the point at which the agent's need for diagnosis arises: When a conflict occurs between an observation and the agent's estimate of the state, the suitability of the agent's original goal and plan are no longer guaranteed, and the agent must evaluate possible alternative goals, decide to select a new goal or retain its original goal, and create a new plan. To ensure that the new plan is based on well-founded assumptions about the current state of the environment, the agent must infer the reason for the expectation failure: it must solve a diagnosis problem by revising its hypothesized execution history and identifying the faulty assumptions about the initial state of the environment. As the agent pro-

[1] Note that, because agent actions are known and environmental events are deterministic, failures in state estimates can be traced back to one or more incorrect assumptions about the initial state.

gresses toward its goal, each time an observation disagrees with its predictions the agent must solve a new diagnosis problem. Thus the overall online diagnosis need for the agent is inherently iterative over the course of execution: at each instance of the diagnosis problem, the agent has access to its previous solution a starting point to reconcile its predictions and observations.

3.2 The Situation Calculus

The situation calculus is a logical language that can be used to create formal definitions for transition systems with states and actions. The basic model is typical of state transition systems: states consist of values assigned to a set of predicate fluents that may be parameterized over a finite set of objects. Actions have preconditions over fluent values, and effects that modify fluent values. Actions cannot occur concurrently.

However, the situation calculus is uniquely well suited to diagnosis problems because its fundamental formalism is not the environmental state. Instead, the focus is the *situation*: the complete sequence of actions that have occurred so far in the system. Because the situation is characterized as an action sequence, it may be concretely reasoned about even when the current state is only partially observable.

The modern situation calculus [Lin and Reiter 1994] has divided into a number of *dialects*, with varying properties and constraints, all expressing models for reasoning about situations as sequences of transitions. For our work, we are using the definitions introduced by McIlraith [1997] (see also [Sohrabi *et al*. 2010]) in her research on explanatory diagnosis problems. We provide a brief introduction to the McIlraith dialect of the situation calculus below.

The situation calculus definition for a specific diagnosis problem can be broken into two main parts: the set of symbols that exist in the problem domain, and the set of axioms that control the interactions of those symbols. We use examples taken from a simplified model of the Autonomous Squad Member (ASM) domain [Gillespie *et al.* 2015], which describes a robotic agent that assists a squad with conducting a surveillance mission.

Situation Calculus Symbols

- **Situations**: The complete sequence of actions (often indicated with the symbol s) that have occurred in the system up to a given point. The null, initial situation is denoted as S_0, and the distinguished function 'do' describes situation transitions: $s_2 = do(a, s_1)$ denotes the situation s_2 resulting from performing action a in situation s_1.
- **Objects** O: A finite set of typed objects that exist in the environment (examples: squad members, trees, the ASM robot itself)
- **Fluents** F: A set of predicates[2] over objects, with values that vary across situations. For this reason, the situation is always the last parameter in a fluent expression. For example, the truth value of the predicate fluent $Wounded(soldier1, s)$ indicates whether $soldier1$ is wounded after the action sequence denoted by situation s. Note that, by itself, s is generally not sufficient to determine the value of a fluent $F(x, s)$; this value is also dependent on the initial system state, T_{S_0}, which we discuss below.
- **Actions** A: There is a finite set of action symbols. The behavior of these actions (i.e., their preconditions and effects) are encoded in the precondition and successor state axioms described below. The atomic expression $Poss(a, s)$ indicates whether action a is possible in situation s (and, as with fluents, the value of $Poss(a, s)$ is partially dependent on the initial system state T_{S0}.)

Situation Calculus Axioms

- **Foundational Axioms** Σ_{found} : The foundational axioms specify the domain-independent framework of the situation calculus, including the definition of situations (described informally above) and the framing axiom or domain closure axiom (described informally below). They also define the predecessor relation $s \sqsubset$ s', which holds if and only if s is a strict prefix of s' (recall that each situation encapsulates an entire action sequence, starting from the initial null situation, S_0)
- **Initial Constraints** T_{SC}^{S0} : A set of constraints on fluents, which all valid initial states must satisfy (for example, $\forall x \in Soldiers: \neg Wounded(x, S_0)$).
- **Successor State Axioms** T_{SS}: This set contains one pair of Successor State Axioms (SSA) for each fluent; it encodes the effects each (possible) action can have on the fluent's value[3]. These are of the form:

  $$F(x_1, \dots, x_n, do(a, s)) \equiv \Phi_F(a, x_1, \dots, x_n, s)$$
  $$\neg F(x_1, \dots, x_n, do(a, s)) \equiv \Phi_{\neg F}(a, x_1, \dots, x_n, s)$$

  where Φ_F is a formula uniform in s (ie, not referring to any predecessors of s), and $a, x_1, \dots, x_n$ are free variables spanning all applicable actions and parameter values for F. For example:

  $Wounded(x, do(a, s)) \equiv [Wounded(x, s) \vee (a = IsShot(x))]$

  $\neg Wounded(x, do(a, s)) \equiv [\neg Wounded(x, s) \vee (a = Treated(x))]$
- **Action Precondition Axioms** T_{AP}: This set contains one precondition axiom for each action symbol in the domain. These are of the form:

  $$Poss(a(x_1, \dots, x_n), s) \equiv \Pi_a(x_1, \dots, x_n, s)$$

  where Π_a is a formula uniform in s which defines all conditions under which a can be performed in s, and $a, x_1, \dots, x_n$ are free variables. For example: $Poss(IsShot(x), s) \equiv [UnderAttack(x) \wedge Exposed(x)]$
- **Unique Action Name Axioms** T_{UNA}: These axioms enforce unique names for actions.

[2] An adaption to functional fluent values exists, but will not be treated in this paper.

[3] The framing axiom, included in Σ_{found}, asserts that fluents do not change value except as specified by the fluent's SSA.

- **Initial State** T_{S_0}: These axioms specify the complete set of initial fluent values for a given instance of the problem. Because situations specify action sequences rather than environmental states, T_{S_0} is necessary (in general) to compute which fluent values hold and which actions are possible in a given situation.

As a whole, the axiomatization of a given problem domain in the situation calculus can be written as:

$$T = \Sigma_{found} \wedge T_{SC}^{S0} \wedge T_{SS} \wedge T_{AP} \wedge T_{UNA} \wedge T_{S_0}$$

An *executable situation* is a situation whose actions are all possible under the axiomatization of the domain (again dependent on T_{S_0}). This has the natural definition, i.e:

$$executable(s) \Leftrightarrow \forall a, s'[(do(a, s') \sqsubseteq s) \supset Poss(a, s')]$$

Additionally, McIlraith introduces the following useful formalisms specific to the explanatory diagnosis problem.

Observation $OBS[S_0, s]$: This encodes the set of all observation information about the system that is available to the process performing the diagnosis algorithm. Formally, it is a sentence of the situation calculus whose only free variable is s, and whose situation terms are restricted to s_i such that $S_0 \sqsubseteq s_i \sqsubseteq s$ (i.e., it refers only to situations that have occurred during the current execution).
Hypothesis $H(S_0)$: This is the diagnostic algorithm's hypothesis regarding the value of the initial state, T_{S_0}.
Diagnosis $[H(S_0), A]$: Given a problem domain axiomatization T and an observation $OBS[S_0, s]$, the diagnosis consists of a hypothesis $H(S_0)$ and a proposed action history (i.e., hypothesized sequence of actions) A, such that the actions A could have occurred starting from the hypothesized initial state $H(S_0)$ without contradicting the observation $OBS[S_0, s]$. Formally:

$$T \cup H(S_0) \models$$
$$\exists s: [(s = do(A, S_0)) \wedge executable(s) \wedge OBS[S_0, s]]$$

3.3 The State-set Framework

The state-set framework introduced by Pang and Holte [2011] provides a mechanism for formally analyzing transitions and sequences of transitions between *sets* of states. While this concept applies in a variety of search problems in planning, we use it specifically to model sets of possible states (i.e., states that are consistent with the current knowledge of the agent). We provide a brief overview of concepts from the state-set framework that are central to our work, tailored to the situation calculus formalisms and our iterative explanatory diagnosis context. We then provide two useful theorems which are implied by the constraints of the explanatory diagnosis problem.

State-set: Let $\mathbb{E}$ be the set of all possible environmental states for a given transition system. Then $E \subseteq \mathbb{E}$ is a *state-set* with respect to $\mathbb{E}$. If $|E| = 1$, (i.e. E contains only one possible state) then we say E represents a *known state*.
State-set Operator: In a state transition system, actions can be considered functions of the form $a: \mathbb{E} \rightarrow \mathbb{E}$ (i.e. an action transitions from one environmental state to the next, $a(e_1) = e_2$). The set of states that satisfy the preconditions of a is referred to as PRE_a, and the set of states that could be the result of taking action a is referred to as $POST_a$ (formally, $POST_a = \{a(e) | e \in PRE_a\}$) For $E \subseteq PRE_a$, we define the *state-set operator* for an action a as: $a(E) = \{a(e) | e \in E\}$.
Action Path: Given an initial state-set I, if a situation $s = a_1, a_2, a_3 \ldots a_n$ is executable for at least one initial state in I, this means that: $[a_1(I) \cap PRE_{a_2} \neq \emptyset] \wedge [a_2(a_1(I)) \cap PRE_{a_3} \neq \emptyset] \ldots \wedge [a_{n-1}(\ldots(I)) \cap PRE_{a_n} \neq \emptyset]$. We'll use the shorthand $s(I)$ to refer to the set $a_n(a_{n-1}(\ldots a_1(I)))$. We say s forms an *action path* from state-set I to state-set $s(I)$, and the states-sets $a_i(\ldots(I))$ are *intermediate state-sets* on this path. Figures 1a-1c provide illustrations of action paths.

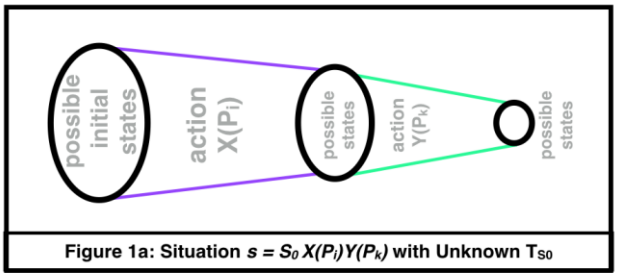

Figure 1a: Situation $s = S_0 X(P_i) Y(P_k)$ with Unknown T_{S0}

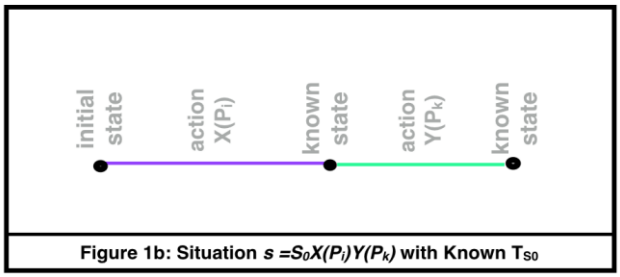

Figure 1b: Situation $s = S_0 X(P_i) Y(P_k)$ with Known T_{S0}

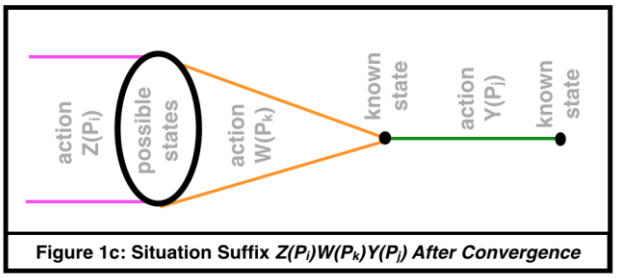

Figure 1c: Situation Suffix $Z(P_i)W(P_k)Y(P_j)$ After Convergence

Figure 1: Action paths in the state-set framework

We now show that the intermediate state sets on an action path decrease in size monotonically (depicted in Figure 1).
Theorem 1:
Claim 1: *If a is an action and E is a state-set such that $PRE_a \supseteq E$, then: $|E| \geq |a(E)|$.*
Claim 2: *Additionally, if a situation s is executable for at least one state in an initial state-set I, then: $|I| \geq |s(I)|$, and for all s', s'' such that $s' \sqsubseteq s'' \sqsubseteq s$, $|s'(I)| \geq |s''(I)|$*
Proof: The first claim is implied by of the fact that a is a function over the individual states in E (intuitively: taking an action in a state will never lead to the agent being in *two*

different states afterwards). The second claim results from an application of the transitive property to the first claim. ■

Recall that the main objective in our iterative explanatory diagnosis problem is to obtain a robust estimate of the current state to supply to the goal reasoning agent. Corollary 1 states that, if explanatory diagnosis ever reaches definite knowledge of the current state, all subsequent states will be definitely known.

Corollary 1: *If s forms an action path from I to s(I), and there exists* $s' \sqsubseteq s$ *such that* $|s'(I)| = 1$ *(i.e. there is an intermediate state-set which is a known state), then* $\forall s''$ *such that* $s' \sqsubseteq s'' \sqsubseteq s$*, we have* $|s''(I)| = 1$ *(all subsequent state-sets on the action path are also known states). In particular, if I is a known state, all state-sets on the path are known states.*

Proof: This is a consequence of Theorem 1. ■

Finally, we note that in the goal reasoning context, periodic observations provide new information that may reduce the set of possible states. However, observations are not state-set operators because they do not transition between states, they merely provide information by which an inference algorithm can eliminate possible sates. We've chosen to depict this using a rectangular bar as in Figure 2. In the following section we discuss in greater detail the role of observations in the solution space for the explanatory diagnosis problem.

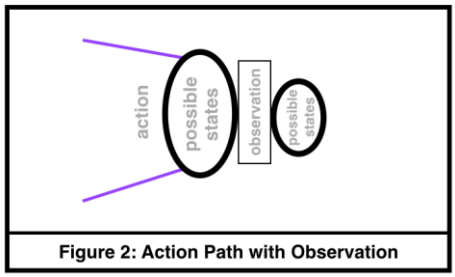

Figure 2: Observations in the state-set framework

4 A Novel Abstraction

In this section, we begin by providing a formal mapping between the structures of our iterative explanatory diagnosis problem model and the framework of the situation calculus. We then present our abstraction of the integrated situation/state-set solution space for the explanatory diagnosis problem, and describe how the explanation search algorithm operates with respect to this solution space.

4.1 Definition in the Situation Calculus Framework

There exists a relatively natural mapping from the iterative explanatory diagnosis problem model into the situation calculus framework. Fluents have similar interpretations in both systems; Both environmental events and agent actions can both be encoded as action symbols in the situation calculus. To address the problem of expressing co-occurring environmental events in the situation calculus (which does not allow concurrent actions), we refer to event sets as defined in §3.1. The fact that event sets are triggered deterministically as soon as their preconditions are satisfied is encoded in the action precondition axioms.

Given an iterative explanatory diagnosis problem as described in §3.1, with a finite set of objects O, observable fluents F_{OBS} and hidden fluents F_{HID} , observable agent actions A_{AG}, unobservable deterministically-triggered environmental event sets A_{EV}, knowledge of valid initial states I_{VAL}, and ground-truth initial state e_{start} , the mapping from this iterative explanatory diagnosis problem into the situation calculus is as follows:

Mapping to Situation Calculus Symbols:

- **Situations**: Sequences of interleaving actions and event-sets, starting in the null initial situation S_0.
- **Objects** O: O
- **Fluents** F: $F_{OBS} \cup F_{HID}$
- **Actions** A: $A_{AG} \cup A_{EV}$

Mapping to Situation Calculus Axioms:

- **Initial Constraints** T_{SC}^{S0} : I_{VAL}
- **Successor State Axioms** T_{SS}: For each fluent in $F_{OBS} \cup F_{HID}$, and for every action or event set in $A_{AG} \cup A_{EV}$ that references that fluent in its effects, a pair of positive and negative SSA's are defined as described in §3.2.
- **Action Precondition Axioms** T_{AP}: For each event set in $a_V \in A_{EV}$, an Action Precondition Axiom Π_{a_V} is defined over the fluents in $F_{OBS} \cup F_{HID}$ that are referenced in the preconditions of a_V, as described in §3.2. We use $\Pi_{EV} = \bigvee_{a_{V \in A_{EV}}} [\Pi_{a_V}]$ to represent the union of all event set preconditions. Recall that in our model, events sets must happen immediately when they are triggered and agent actions cannot occur until the last triggered event sequence has completed. To encode this constraint, the Precondition Axiom Π_{a_G} for each agent action in $a_G \in A_{AG}$ include both $\neg\Pi_{EV}$ in addition to the fluents in F_{OBS} that are referenced in the action's precondition (the preconditions for agent actions are always observable).
- **Unique Action Name Axioms** T_{UNA}: Defined for each action symbol.
- **Initial State Value** T_{S_0}**:** e_{start}

The iterative explanatory diagnosis problem is distinct from the diagnosis formalism previously defined in §3.2, in that it occurs iteratively in online (e.g., goal reasoning) agents: At *all* times the goal reasoning agent maintains a hypothesis about the initial state value $H(S_0)$ and an explanation (a consistent, estimated execution history A). Each time the agent encounters an unexpected observation, it searches for a diagnosis which will consist of a revised hypothesis $H'(S_0)$ and revised execution history A', preceding from the previous $[H(S_0), A]$.

Valid diagnoses must be consistent with the agent's (iteratively updated) observation, which is comprised of all

agent actions and the values of F_{OBS} in the situations that immediately preceded those actions. We introduce notation to formally define the observation as a sentence in the situation calculus. We refer to the sequence of agent actions that have occurred so far in situation s as $A_{agent}(s) = a_1, a_2, \ldots a_n$. For each $a_i \in A_{agent}(s)$ we use s_{a_i} to refer to the situation immediately *before* action a_i occurred (i.e., the situation after a_i is $do(a_i, s_{a_i})$). We further use the notation $F_{OBS+}(s_i) = \bigwedge_{F_i \in F, X \in O^k : [F_i(X,s_i)=T]} [F_i(X, s_i)]$ and $F_{OBS-}(s_i) = \bigwedge_{F_i \in F, X \in O^k : [F_i(X,s_i)=F]} \neg[F_i(X, s_i)]$ to indicate the true and false grounded observable fluents in situation s_i. We then have:

$$OBS[S_0, s] = \forall a_{i \in A_{agent}(s)} : [do(a_i, s_{a_i}) \sqsubseteq do(a_{i+1}, s_{a+1})] \wedge [F_{OBS+}(s_{a_i}) \wedge F_{OBS-}(s_{a_i})]$$

4.2 The Situation/State-set solution space

The space of valid solutions to the iterative explanatory diagnosis problem consists of two interdependent sub-spaces: possible states and possible situations. There is the space of possible environmental states at each step in the execution, and there is the set of possible situations (consisting of the known agent actions and possible environmental event sequences). Because events occur deterministically, information that eliminates possible states also reduces the set of possible event sequences. Similarly, refining the set of possible event sequences reduces the set of possible current states. For an agent's hypothesized execution history and estimated state to be valid, they must lie within this solution space.

Figure 3 applies the combined situation/state-set abstraction to illustrate the evolution of this solution space, through the first action and observation, for an agent whose initial plan begins with the actions: $[action1, action2, action3]$. Figure 3a depicts the solution space before the first action is taken. The set of possible initial states is constrained by the agent's knowledge of the starting state, taken from an initial observation and the set of constraints that determine which initial states are valid, $(F_{OBS}(S_0) \wedge I_{VAL})$.

The set of possible states following the first action will be smaller if the action taken is not a one-to-one function. For example: taking the action $MoveForward(robot1, S_0)$ would have the effect of mapping the possible initial states where the fluent $\neg Moving(robot1, S_0)$ holds, *and* the possible initial states where $Moving(robot1, S_0)$ holds, to states where fluent $Moving(robot1, s_1)$ is guaranteed to hold. This reduces the set of possible states after the action—and *overwrites* the initial state information $[\neg]Moving(robot1, S_0)$, such that it may be impossible to infer with certainty from later observations. (One implication of this reduction in state-set size is that it is possible to reach a known intermediate state on an action path, and thus be guaranteed to satisfy the agent's diagnosis needs by correctly predicting all future states and event sequences, *without ever* being able to correctly determine with certainty the states and situations that preceded the known state.)

Dependent on the state following the first action, there are many event sequences that could possibly occur, (the space of possible event sequences is depicted in grey in Figure 3 Panel I). Some event sequences will affect observable fluents differently than others, and will result in different observation values following the sequences. Making an observation thus eliminates possible situations and reduces the solution space. Figure 3 Panel II depicts the evolution of the solution space after the first observation is taken, and Figure 3 Panel III shows the solution space after the second observation.

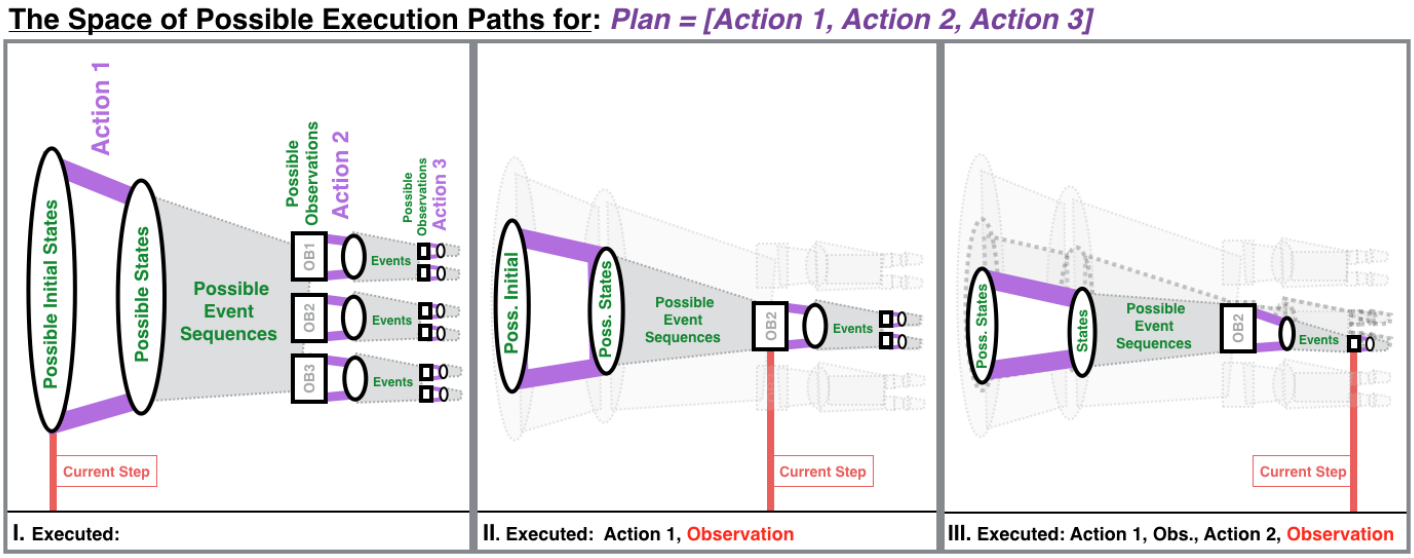

Figure 3: The situation/state-set solution space

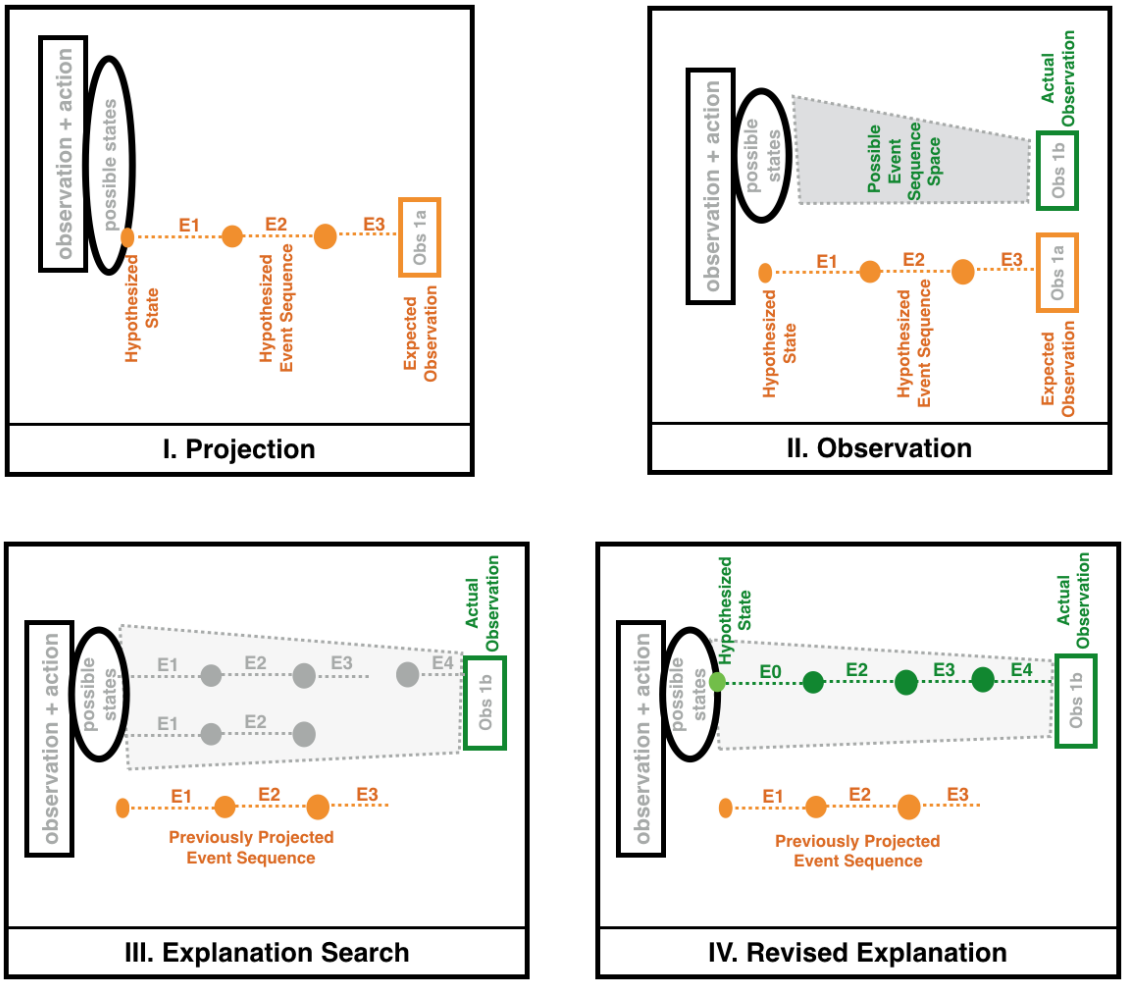

Figure 4: Four steps of online planning with explanatory diagnosis

4.3 Depiction of Explanatory Diagnosis in an Online Agent

We now apply our abstraction to discuss how the explanation search algorithm [Molineaux and Aha 2015] iteratively navigates the solution space to identify a revised, consistent explanatory diagnosis ('*explanation*') each time an observation conflicts with the agent's expectations. Figure 4 illustrates four steps of an online planning agent with explanatory diagnosis over the subsequence of the situation occurring between the most recent two observations (the complete explanatory diagnosis search continues revisions back to correct the hypothesized initial state). The steps of the explanatory search algorithm proceed as follows: (1) Initially, the agent maintains a hypothesized state and projects from there the expected event sequence and observation (Panel I). (2) The expected state is compared to the observation and discrepancies, if any, are identified (if there exists one or more discrepancies, the space of possible event sequences no longer contains the previously hypothesized event sequence (Panel II)). (3) The agent begins explanation revision by considering *all possible* edits to the previously hypothesized execution history (e.g., adding or removing events) that might affect the discrepancy regarding the unexpected observed fluent value (Panel III). These edits to the explanation may introduce additional discrepancies (such as unsatisfied preconditions or conflicts with previous observations), causing the edited explanation to violate the space of possible situations. (4) The discrepancy-resolution process (searching across all applicable edits) is recursively applied, until a resulting explanation has no further discrepancies. One or more resolutions must change a hypothesized initial state value (because, in a deterministic environment, no other execution history can have occurred with the same hypothesized initial state). The revised explanation must lie entirely inside the set of possible situations, implying that it is consistent with the domain constraints defined in §4.1 and is consistent with all observations (Panel IV).

5 Discussion and Future Work

Our goal has been to formalize and clarify the iterative explanatory diagnosis problem for goal reasoning agents in partially observable environments. Here we highlight some initial insights, with the objective of demonstrating concrete applications of these insights in our future work.

We start by noting that techniques which use an explicit estimation of the situation/state-set solution space to limit

the scope of the diagnosis search could help improve search efficiency.

There are several techniques that might be effective for guiding the explanation search to stay within the set of possible states/situations as much as possible, reducing time spent on fruitless branches in the search. Case-based approaches that store event sequences associated with common classes of fluent observation values might be applicable for guiding search in this fashion. Finally, fluent dependency data structures (mapping each expected observation fluent to previous estimated hidden fluents hypothesized to affect its value) could allow us to infer likely candidates for revised hypothesized states, directly from the unexpected observation.

Another way to limit the scope of explanation search is temporally: Since the agent's primary diagnosis need is to attain a robust estimate of its current state, rather than a complete execution history, it may not be practically useful to revise explanations completely back to the initial state (which, as we established in §4.2, will in many cases be impossible to identify with certainty.) The agent could use a method for estimating the size and structure of intermediate state-sets to construct a heuristic identifying when significant information loss has occurred between state-sets. This would be useful for selecting a point to terminate explanation revision, accepting uncertainty about the history of fluents whose true values may not be inferable previous to that point. This approach could be exploited in conjunction with an agent using diagnostic planning to intentionally select actions that reduce the set of possible current states.

Additionally, the true size and complexity of the solution space depicted in Figure 3 depends on the structure of the fluent set (which determines the size and internal structure of possible event-sets) and the design of the action/event preconditions and effects (which affect the space of possible event-sequences, and the reduction in the size of the state-set after each action). Since different representations of a given domain may result in varying levels of complexity in the modeled solution space, which in turn causes varying degrees of efficiency in explanation search, future work could investigate the domain-representation qualities that affect efficiency of explanatory diagnosis search.

Finally, in the future we may consider actions with non-deterministic effects, or non-deterministically triggered events, and adjust our abstraction to represent these.

Acknowledgements

Thanks to OSD ASD (R&E) for sponsoring this research.

References

[Abelson *et al*. 1985] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs.* Cambridge, MA: MIT Press, 1985.

[Baumgartner *et al*. 2001] Robert Baumgartner, Georg Gottlob, and Sergio Flesca. Visual information extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Databases*. Rome, Italy: Morgan Kaufmann, 2001.

[De Kleer 1986] Joahn De Kleer. An assumption-based TMS. In *Artificial Intelligence*, 1986

[Gillespie *et al*. 2015] Kellen Gillespie, Matthew Molineaux, Michael W. Floyd, Swaroop S. Vattam, and David W. Aha. Goal reasoning for an autonomous squad member. In *Goal Reasonning: Papers from the ACS Workshop*. Atlanta, GA: Georgia Institute of Technology, Institute for Robotics and Intelligent Machines, 2015.

[Kalman 1960] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. In *Journal of Basic Engineering*, 1960.

[Lin and Reiter 1994] Fangzhen Lin and Raymond Reiter. State constraints revisited. *Journal of Logic and Computation*, 4(5):655-678. Oxford University Press, 1994.

[McIlraith 1997] Sheila McIlraith. Towards a formal account of diagnostic problem solving. PhD thesis. Toronto, Ontario, Canada: University of Toronto, Department of Computer Science, 1997.

[Sohrabi *et al*. 2010] Shirin Sohrabi, Jorge A. Baier, and Sheila A. McIlraith. Diagnosis as planning revisited. In *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*. Toronto, Ontario, Canada: AAAI Press, 2010.

[Molineaux and Aha 2015] Matthew Molineaux and David W. Aha. Continuous explanation generation in a multi-agent domain. In *Proceedings of the Third Conference on Advances in Cognitive Systems*. Atlanta, GA: Cognitive Systems Foundation, 2015.

[Molineaux and Aha 2014] Matthew Molineaux and David W. Aha. Learning unknown event models. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. Quebec City, Quebec, Canada: AAAI Press, 2014.

[Molineaux *et al*. 2010] Matthew Molineaux, Matthew Klenk, and David W. Aha. Goal-driven autonomy in a Navy strategy simulation. In *Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press, 2010.

[Pang and Holte 2011] Bo Pang and Robert C. Holte. State-set search. In *Fourth Annual Symposium on Combinatorial Search*, 2011.

[Thrun *et al*. 2005] Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. Probabilistic Robotics. Cambridge, MA: MIT Press, 2005.

[Vattam *et al*. 2013] Swaroop S. Vattam, Matthew Klenk, Matthew Molineaux, and David W. Aha. Breadth of approaches to goal reasoning: A research survey. In *Goal Reasoning: Papers from the ACS Workshop*. College Park, MD: University of Maryland, Department of Computer Science, 2013.