Formulating diagnostic problem solving using an action language with narratives and sensing

Chitta Baral
Department of CSE
Arizona State University
Tempe, AZ 85287
chitta@asu.edu

Sheila McIlraith
Knowledge Systems Lab
Stanford University
Stanford, CA 94305
sam@ksl.stanford.edu

Tran Cao Son
CS Department
Univ. of Texas at El Paso
El Paso, Texas 79968
tson@cs.utep.edu

May 22, 2001

Abstract

Given a system and unexpected observations about the system, a diagnosis is often viewed as a fault assignment to the various components of the system that is consistent with (or that explains) the observations. If the observations occur over time, and if we allow the occurrence of (deliberate) actions and (exogenous) events, then the traditional notion of a candidate diagnosis must be modified to consider the *possible occurrence of actions* and events that could account for the unexpected system behavior.

In the presence of multiple candidate diagnoses, we may need to perform actions and observe their impact on the system, to be able to narrow the list of possible diagnoses, and possibly even initiate some repair. A plan that guarantees such narrowing will be referred to as a *diagnostic plan*, and if this plan also guarantees that at the end of the execution of the plan, the system has no faults then we refer to it as a *repair plan*.

Since actions and narrative play a central role in diagnostic problem solving, we characterize diagnosis, diagnostic planning and repair with respect to the existing action language $\mathcal{L}$, extended to include static constraints, sensing actions, and the notion of observable fluents. This language is used to provide a uniform account of diagnostic problem solving. We also show that our formulation expands on the earlier formulation of diagnosis with respect to a single state, and the notion of testing.

1 Introduction

Consider the following narrative involving diagnosis.

> John gets up in the morning. He turns on the switch of his lamp, and reads the morning newspaper. He then turns off the switch and does other things before going to work. After he gets home from work, he enters his room and turns on the switch of his lamp again. This time, *the lamp does not turn on.* John thinks that maybe either the bulb is broken, or the switch of the lamp is broken, or the power cord is broken, or there is no power at the outlet. He does nothing about it and goes to his bathroom and turns on the light switch, observing that even that light does not turn on. He thinks perhaps there is no power at home, but then he notices that his electric clock is working, so he figures that there is power in at least part of his home. Now he is worried and goes to

his garage to check his fuse box and finds that one of the fuses is blown. He replaces that fuse and comes back to his room. He turns on his lamp switch and voila it works.

This narrative illustrates the process of diagnostic problem solving. In particular it illustrates that diagnostic problem solving must involve reasoning about the evolution of a dynamical system. Triggered by an observation of system behavior that is inconsistent with expected behavior – in this case, the fact that when John turned on the lamp it did not emit light, diagnostic problem solving involves:

• generating candidate diagnoses based on an incomplete history of events that have occurred and observations that have been made.

• in the event of multiple candidate diagnoses, performing actions to enable observations that will discriminate candidate diagnoses. The selection of a particular action is often biased towards confirming the most likely diagnosis, or the one that is easiest to test.

• generating (possibly with conditional) plans, comprising both world-altering actions and sensing actions, to discriminate candidate diagnoses.

• updating the space of diagnoses in the face of changes in the state of the world, and in the face of new observations.

The long-term objective of our work is to develop a knowledge representation and reasoning capability that emulates diagnostic problem solving processes such as John's. Following [McI97b], we argue that such a comprehensive account of diagnostic problem solving must involve reasoning about action and change. In this paper we augment and extend the work of [McI97a, McI98, McI97b] in several important ways. The main contributions of this paper are:

• We define diagnosis with respect to a narrative.

• We define the notions of diagnostic and repair planning, within a language that integrates sensing actions and world-altering actions. Thus, we are able to distinguish between changes in the state of the world, and changes in an agent's state of knowledge.

• In support of this endeavor, we extend the action language $\mathcal{L}$ to support static causal laws, sensing actions and the notion of observable fluents. $\mathcal{L}$ was originally developed to support narratives (e.g., [MS94, Pin94]).

None of the above issues have been explored either in the model-based diagnosis literature or in the reasoning about action literature. Also notable is that unlike most other accounts of diagnosis, our account allows nondeterministic effects of actions and sensing actions. Finally, our work is distinguished from most previous work in defining diagnosis in terms of a diagnostic model, rather than in terms of failing components and/or actions sequences. Our focus in this paper is to develop an appropriate formulation for diagnostic problem solving. Implementation and complexity issues will be discussed in a future work.

The paper is organized as follows. In Section 2 we provide an overview of the language $\mathcal{L}$ with the addition of static causal laws. In Section 3 we use the extended language to define when we may need to do a diagnosis and what a diagnosis is with respect to a narrative. In Section 4 we further extend our action language to allow sensing actions and to accommodate the distinction between an observable fluent and an unobservable fluent. We then use this language to define the notion of a conditional plan, and the related notions of diagnostic and repair planning. Finally, in Section 6 we summarize and discuss related work.

2 Specifying narrative in $\mathcal{L}$

The propositional language $\mathcal{L}$ was developed in [BGP97, BGP98] to specify narratives and to reason with them. The ontology of $\mathcal{L}$ assumes a single agent's world which evolves in a branching tree from an initial situation s_0. The branches of the tree are determined by the actions that can be performed in a situation – by either the agent or the environment – to transition the world to a new situation. Thus, each situation is simply an action history from s_0. It is also assumed that actions cannot happen simultaneously and no actions occur except those needed to explain the facts of the domain. The distinguished situation s_c refers to the current situation. Properties or relations that change from situation to situation are called *fluents*. Since the language is propositional, there is a finite number of states of the system, each describing what fluents are true in that state. The language $\mathcal{L}$ defined in this section differs from the original language $\mathcal{L}$ in that it allows nondeterministic effects of actions. In this section we extend $\mathcal{L}$ with static causal laws. In Section 4.1, we extend our language further with sensing actions, and observables.

In this paper, we will describe the main aspects of the language $\mathcal{L}$ by dividing it into three components: a domain description language $\mathcal{L}_D$, a language to specify observations $\mathcal{L}_O$, and a query language $\mathcal{L}_Q$.

2.1 $\mathcal{L}_D$: The domain description language

The alphabet of $\mathcal{L}_D$ – a language that closely follows the language $\mathcal{AC}$ from [Tur97] – comprises two nonempty disjoint sets of symbols: the set of fluents $\mathbf{F}$, and the set of actions, $\mathbf{A}$. A *fluent literal* (or *literal*) is a fluent or a fluent preceded by $\neg$. A *fluent formula* is a propositional formula constructed from literals. Propositions in $\mathcal{L}_D$ are of the following forms:

$$a \textbf{ causes } \varphi \textbf{ if } \psi \tag{1}$$

$$\varphi \textbf{ if } \psi \tag{2}$$

$$\textbf{impossible } a \textbf{ if } \psi \tag{3}$$

where a is an action, and φ, and ψ are fluent formulas.

Propositions of the form (1) describe the direct effects of actions on the world and are called *dynamic causal laws*. Propositions of the form (2), called *static causal laws*, describe causal relation between fluents in a world. Propositions of the form (3), called *executability conditions*, state when actions are not executable.

A *domain description* D is a set of propositions in $\mathcal{L}_D$.

The main difference between $\mathcal{L}_D$ and the action description part of $\mathcal{L}$ [BGP97, BGP98] is the presence of static causal laws in $\mathcal{L}_D$, which are critical for representing the behavior of the device being diagnosed. Intuitively, a static causal law of the form (2) states that whenever ψ holds (i.e. evaluates to true wrt. the world) φ must also hold. A static causal law, whose right hand side (i.e., ψ) is equivalent to *true*, represents a relation which always holds. For example, ϕ **if** *true* expresses that ϕ holds in every state of the domain.

A domain description given in $\mathcal{L}_D$ defines a transition function from actions and states to a set of states. (Recall, actions may be nondeterministic.) Intuitively, given an action, a and a state, s the transition function $\Phi(a, s)$ defines the set of states that may be reached after executing the action

a in state s. If $\Phi(a, s)$ is an empty set it means that a is not executable in s. We now formally define this transition function.

Let D be a domain description in the language of $\mathcal{L}_D$. An *interpretation* I of the fluents in $\mathcal{L}_D$ is a maximal consistent set of fluent literals drawn from $\mathbf{F}$. A fluent f is said to be true (resp. false) in I iff $f \in I$ (resp. $\neg f \in I$). The truth value of a fluent formula in I is defined recursively over the propositional connective in the usual way. For example, $f \wedge q$ is true in I iff f is true in I and q is true in I. We say that φ holds in I (or I satisfies φ), denoted by $I \models \varphi$, if φ is true in I.

A set of formulas from $\mathcal{L}_D$ is *logically closed* if it is closed under propositional logic (wrt. $\mathcal{L}_D$).

Let V be a set of formulas and K be a set of static causal laws of the form φ **if** ψ. We say that V is closed under K if for every rule φ **if** ψ in K, if ψ belongs to V then so does φ. By $Cn(V \cup K)$ we denote[1] the least logically closed set of formulas from $\mathcal{L}_D$ that contains V and is also closed under K.

A *state* of D is an interpretation that is closed under the set of static causal laws of D.

An action a is *prohibited (not executable)* in a state s if there exists an executability condition of the form

$$\textbf{impossible } a \textbf{ if } \varphi$$

in D such that φ holds in s.

The *effect of an action* a in a state s of D is the set of formulas $e_a(s) = \{\varphi \mid D$ contains a law a **causes** φ **if** ψ and ψ holds in $s\}$.

Given the domain description D containing a set of static causal laws R, we formally define $\Phi(a, s)$, the set of states that may be reached by executing a in s as follows.

1. If a is not prohibited (i.e., executable) in s, then

$$\Phi(a, s) = \{s' \mid Cn(s') = Cn((s \cap s') \cup e_a(s) \cup R)\};$$

2. If a is prohibited (i.e., not executable) in s, then $\Phi(a, s)$ is $\emptyset$.

The intuition behind the above formulation is as follows. The direct effects of an action a in a state s are determined by the dynamic causal laws and are given by $e_a(s)$. All formulas in $e_a(s)$ must hold in any resulting state. In addition, the static causal laws, R determine additional formulas that must hold in the resulting state. While the resulting state should satisfy these formulas, it must also be otherwise closed to s. These three aspects are captured by the definition above. Observe that when R is empty and a is executable in state s, that $\Phi(a, s)$ is equivalent to the set of states that satisfy $e_a(s)$ and are closest to s using the symmetric difference[2] as the measure of closeness [MT95]. Additional explanation and a description of the motivation behind the above definition can be found in [Bar95, MT95, Tur97].

The situation calculus formulations in [Rei91, LR94, Rei98, McI97b] also provide definitions for $\Phi(a, s)$. We believe that the formulation here – being at a high level – is more compact in the sense that we can formulate it without introducing a lot of additional background formulations.

Every domain description D in a language $\mathcal{L}_D$ has a unique transition function Φ, and we say Φ is the transition function of D.

[1] Note that a fluent formula φ can be equivalently represented as a static causal law φ **if** *true*.

[2] We say s_1 is strictly closer to s than s_2 if $s_1 \setminus s \cup s \setminus s_1 \subset s_2 \setminus s \cup s \setminus s_2$.

2.2 $\mathcal{L}_O$: The observation language

We assume the existence of a set of situation constants **S** which contains two special situation constants s_0 and s_c denoting the initial situation and the current situation, respectively. Note that *situations* written as s (possibly with subscripts) are different from *states* which are written as s (possibly with subscripts). As with the situation calculus, the ontology of our language differentiates between a situation, which is a history of the actions from the initial situation, and a state, which is the truth value of fluents at a particular situation.

Observations in $\mathcal{L}_O$ are propositions of the following forms:

$$\varphi \ \textbf{at} \ \mathsf{s} \tag{4}$$

$$\alpha \ \textbf{between} \ \mathsf{s}_1, \mathsf{s}_2 \tag{5}$$

$$\alpha \ \textbf{occurs\_at} \ \mathsf{s} \tag{6}$$

$$\mathsf{s}_1 \ \textbf{precedes} \ \mathsf{s}_2 \tag{7}$$

where φ is a fluent formula, α is a (possibly empty) sequence of actions, and $\mathsf{s}, \mathsf{s}_1, \mathsf{s}_2$ are situation constants which differ from s_c. (Since the world can be changed without the agent's knowledge, we do not allow the agent to have observations about s_c.)

Observations of the forms (4) and (7) are called *fluent facts* and *precedence facts*, respectively. Observations of the forms (5) and (6) are referred to as *occurrence facts*. These two types of observations are different in that (5) states exactly what happened between two situations s_1 and s_2, whereas (6) only says what occurred in the situation s.

2.3 Narratives

A *narrative* is a pair (D, Γ) where D is a domain description and Γ is a set of observations of the form (4)-(7).

Example 1 Consider a simplified version of the introductory example where we have actions *turn_on, turn_off* and *break(bulb)*. Intuitively, $break(bulb)$ is an exogenous action that can make the bulb defective. We denote the defectiveness of the bulb by the fluent $ab(bulb)$. The information related to John's lamp in the story can then be described by the following narrative $N_0 = (D_0, \Gamma_0)$:

$$D_0 = \begin{cases} \text{(r1)} & turn\_on \ \textbf{causes} \ light\_on \ \textbf{if} \ \neg ab(bulb) \\ \text{(r2)} & turn\_off \ \textbf{causes} \ \neg light\_on \\ \text{(r3)} & \neg light\_on \ \textbf{if} \ ab(bulb) \\ \text{(r4)} & break(bulb) \ \textbf{causes} \ ab(bulb) \\ \text{(r5)} & \textbf{impossible} \ break(bulb) \ \textbf{if} \ ab(bulb) \end{cases}$$

$$\Gamma_0 = \begin{cases} (o1) & \mathit{turn\_on} \ \textbf{occurs\_at} \ \mathsf{s}_0 \\ (o2) & \mathit{turn\_off} \ \textbf{occurs\_at} \ \mathsf{s}_1 \\ (o3) & \mathit{turn\_on} \ \textbf{occurs\_at} \ \mathsf{s}_2 \\ (o4) & \mathsf{s}_0 \ \textbf{precedes} \ \mathsf{s}_1 \\ (o5) & \mathsf{s}_1 \ \textbf{precedes} \ \mathsf{s}_2 \\ (o6) & \mathsf{s}_2 \ \textbf{precedes} \ \mathsf{s}_3 \\ (o7) & \neg \mathit{light\_on} \ \textbf{at} \ \mathsf{s}_0 \\ (o8) & \mathit{light\_on} \ \textbf{at} \ \mathsf{s}_1 \\ (o9) & \neg \mathit{light\_on} \ \textbf{at} \ \mathsf{s}_2 \\ (o10) & \neg \mathit{light\_on} \ \textbf{at} \ \mathsf{s}_3 \end{cases}$$

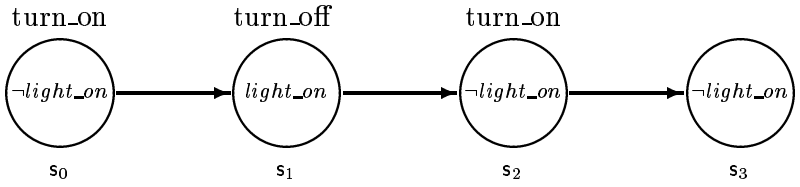


Figure 1: N_0 in picture

Observations are interpreted with respect to a domain description. While a domain description defines a transition function that characterizes what states *may* be reached when an action is executed in a state, a narrative consisting of a domain description together with a set of observations defines the possible situation histories of the system. This characterization is achieved by two functions, Σ and Ψ. While Σ maps situation constants to action sequences, Ψ picks one among the various transitions given by $\Phi(a, s)$ and maps action sequences to a unique state with the condition that $\Psi(\alpha \circ a) \in \Phi(a, \Psi(\alpha))$.

More formally, let (D, Γ) be a narrative. A *causal interpretation* of (D, Γ) is a partial function from action sequences to interpretations of $Lang(\mathbf{F})$, whose domain is nonempty and prefix-closed[3]. By $Dom(\Psi)$ we denote the domain of a causal interpretation Ψ. Notice that $[] \in Dom(\Psi)$ for every causal interpretation Ψ, where $[]$ is the empty sequence of actions.

A *causal model* of D is a causal interpretation Ψ such that:

(i) $\Psi([])$ is a state of D; and

(ii) for every $\alpha \circ a \in Dom(\Psi)$, $\Psi(\alpha \circ a) \in \Phi(a, \Psi(\alpha))$.

A *situation assignment* of $\mathbf{S}$ with respect to D is a mapping Σ from $\mathbf{S}$ into the set of action sequences of D that satisfy the following properties:

(i) $\Sigma(\mathsf{s}_0) = []$;

[3]A set X of action sequences is prefix-closed if for every sequence $\alpha \in X$, every prefix of α is also in X.

(ii) for every $\mathsf{s} \in \mathbf{S}$, $\Sigma(\mathsf{s})$ is a prefix of $\Sigma(\mathsf{s_c})$.

An *interpretation* M of (D, Γ) is a pair (Ψ, Σ), where Ψ is a causal model of D, Σ is a situation assignment of $\mathbf{S}$, and $\Sigma(\mathsf{s_c})$ belongs to the domain of Ψ. For an interpretation $M = (\Psi, \Sigma)$ of (D, Γ):

(i) α **occurs_at** s is true in M if the sequence $\Sigma(\mathsf{s}) \circ \alpha$ is a prefix of $\Sigma(\mathsf{s_c})$;

(ii) α **between** $\mathsf{s}_1, \mathsf{s}_2$ is true in M if $\Sigma(\mathsf{s}_1) \circ \alpha = \Sigma(\mathsf{s}_2)$;

(iii) φ **at** s is true in M if φ holds in $\Psi(\Sigma(\mathsf{s}))$;

(iv) s_1 **precedes** s_2 is true in M if $\Sigma(\mathsf{s}_1)$ is a prefix of $\Sigma(\mathsf{s}_2)$.

An interpretation $M = (\Psi, \Sigma)$ is a *model* of a narrative (D, Γ) if:

(i) facts in Γ are true in M;

(ii) there is no other interpretation $M' = (\Psi, \Sigma')$ such that M' satisfies condition i) above and $\Sigma'(\mathsf{s_c})$ is a subsequence of $\Sigma(\mathsf{s_c})$.

Observe that these models are minimal in the sense that they exclude extraneous actions.

A narrative is *consistent* if it has a model. Otherwise, it is *inconsistent.*

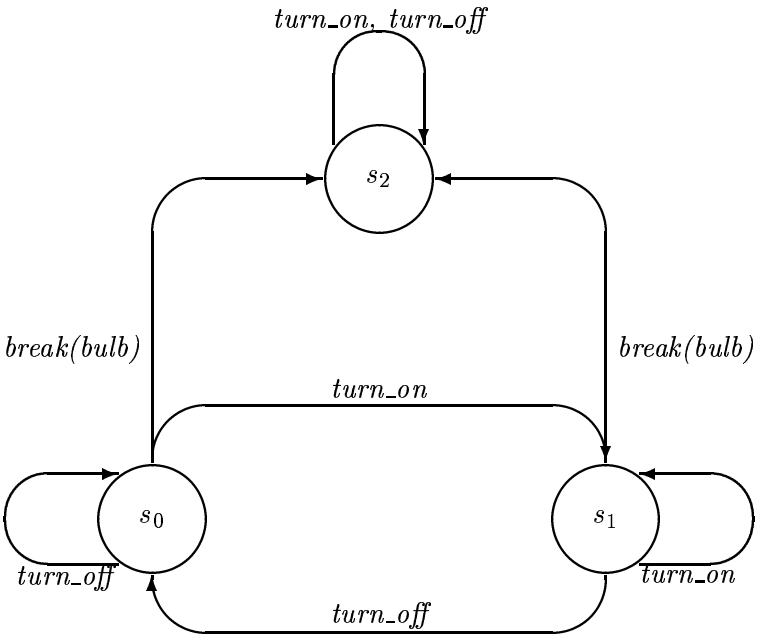
Example 2 Consider the narrative $N_0 = (D_0, \Gamma_0)$ from Example 1. There are four interpretations for the set of fluent formulas of D_0:

$I_0 = \emptyset$, $I_1 = \{light\_on\}$, $I_2 = \{ab(bulb)\}$, and $I_3 = \{ab(bulb), light\_on\}$.

Since $Cn(I_3)$ is not closed under the static causal law (r3) of D_0 ($ab(bulb) \in Cn(I_3)$ and (r3) imply that $light\_on \notin Cn(I_3)$), I_3 is not a state of D_0. Thus, D_0 has only three states $s_0 = \emptyset$, $s_1 = \{light\_on\}$, and $s_2 = \{ab(bulb)\}$. The transition function of D_0 is given by

$$\begin{array}{lll} \Phi(turn\_on, s_0) = \{s_1\} & \Phi(turn\_on, s_1) = \{s_1\} & \Phi(turn\_on, s_2) = \{s_2\} \\ \Phi(turn\_off, s_0) = \{s_0\} & \Phi(turn\_off, s_1) = \{s_0\} & \Phi(turn\_off, s_2) = \{s_2\} \\ \Phi(break(bulb), s_0) = \{s_2\} & \Phi(break(bulb), s_1) = \{s_2\} & \Phi(break(bulb), s_2) = \emptyset \end{array}$$

Φ can be represented graphically as follows.



N_0 has three models $M_1 = (\Psi_1, \Sigma_1)$, $M_2 = (\Psi_2, \Sigma_2)$, and $M_3 = (\Psi_3, \Sigma_3)$, where $\Psi_1([]) = \Psi_2([]) = \Psi_3([]) = s_0$, and

$\Sigma_1(\mathsf{s}_0) = []$,
$\Sigma_1(\mathsf{s}_1) = turn\_on$,
$\Sigma_1(\mathsf{s}_2) = turn\_on \circ turn\_off$, and
$\Sigma_1(\mathsf{s}_3) = \Sigma_1(\mathsf{s}_C) = turn\_on \circ turn\_off \circ turn\_on \circ turn\_off$,

$\Sigma_2(\mathsf{s}_0) = []$,
$\Sigma_2(\mathsf{s}_1) = turn\_on$,
$\Sigma_2(\mathsf{s}_2) = turn\_on \circ turn\_off \circ break(bulb)$, and
$\Sigma_2(\mathsf{s}_3) = \Sigma_2(\mathsf{s}_C) = turn\_on \circ turn\_off \circ break(bulb) \circ turn\_on$.

$\Sigma_3(\mathsf{s}_0) = []$,
$\Sigma_3(\mathsf{s}_1) = turn\_on$,
$\Sigma_3(\mathsf{s}_2) = turn\_on \circ turn\_off$, and
$\Sigma_3(\mathsf{s}_3) = \Sigma_3(\mathsf{s}_C) = turn\_on \circ turn\_off \circ turn\_on \circ break(bulb)$.

Notice the difference between M_1 and M_2. In M_1, the unobserved action ($turn\_off$) occurs after the last observed action whereas in M_2 the unobserved action ($break(bulb)$) occurs prior to it. □

2.4 $\mathcal{L}_Q$: The query language

Queries in $\mathcal{L}_Q$ are of the following form:

$$\varphi \textbf{ after } \alpha \textbf{ at } \mathsf{s} \tag{8}$$

When α in (8) is an empty sequence of actions, and s is the current situation $\mathsf{s_c}$, we often use the notation **currently** φ as a simplification of (8).

A query of the form φ **after** α **at** s is true in a model $M = (\Psi, \Sigma)$ of a narrative (D, Γ), denoted by $(D, \Gamma) \models_M q$, if φ is true in $\Psi(\Sigma(\mathsf{s}) \circ \alpha)$.

A query q is entailed by a narrative (D, Γ), denoted by $(D, \Gamma) \models q$, if q is true in every model of (D, Γ).

3 Diagnosis with respect to narratives

We are now ready to formulate the notion of diagnosis with respect to a narrative. The representation of the system to be diagnosed comprises of static causal laws that describe the behavior of the system itself, as well as the description of the effects of actions on system state, and observations about action occurrences and fluent values over the evolution of the system. We follow the diagnosis literature (e.g., [dKMR92]) and assume that the system is composed of a distinguished set of components that can malfunction. Associated with each component c, is the distinguished fluent $ab(c)$, denoting that the component c is abnormal or broken. Also associated with each component is the distinguished action $break(c)$, a wildcard action which may be used to explain unexpected observations about $ab(c)$. Note that the representation of the system is likely to contain other actions and static causal laws that affect the truth of $ab(c)$. Building on the established diagnosis notation:

Definition 1 (System) A *system Sys* is a tuple $(SD, COMPS, OBS)$ where

$COMPS = \{c_1, \ldots, c_n\}$ is a finite set of components.

SD is a domain description characterizing the behavior of the system, and augmented with dynamic laws of the form $break(c)$ **causes** $ab(c)$, for each component c in $COMPS$.

Given SD, by SD_{ab}, we denote the subset of SD consisting of static causal laws of the form "ψ **if** φ" and dynamic laws of the form "a **causes** ψ **if** φ", where ψ contains $ab(c)$ for some component c.

OBS is a collection of observations about the system.

We define

$$\mathcal{D}(\Delta_1, \Delta_2) = \{\neg ab(c) \ \textbf{at} \ \mathsf{s}_0 \mid c \in \Delta_1\} \cup \{ab(c) \ \textbf{at} \ \mathsf{s}_0 \mid c \in \Delta_2\}.$$

where Δ_1 and Δ_2 are two disjoint sets of components, i.e., $\Delta_1, \Delta_2 \subseteq COMPS$ and $\Delta_1 \cap \Delta_2 = \emptyset$. Intuitively, $\mathcal{D}(\Delta_1, \Delta_2)$ states the status of components belonging to $\Delta_1 \cup \Delta_2$ in the initial situation: components in Δ_1 are okay but those in Δ_2 are broken. For convenience, we write $OK_0 = \mathcal{D}(COMPS, \emptyset)$.

Example 3 Consider a slight variation of the story in our introduction. Assume that the only breakable component of the domain is the *bulb*. Furthermore, assume that John observed that the light is off *immediately* after he turned on the lamp when coming back from work. The story can then be described by a system description $Sys_1 = (SD_1, \{bulb\}, OBS_1)$ where $SD_1 = D_0$ - the domain description defined in Example 1, and OBS_1 as given below.

$$OBS_1 = \begin{cases} \text{(o1')} & \mathit{turn\_on} \textbf{ occurs\_at } \mathsf{s}_0 \\ \text{(o2')} & \mathit{turn\_off} \textbf{ occurs\_at } \mathsf{s}_1 \\ \text{(o3')} & \mathit{turn\_on} \textbf{ between } \mathsf{s}_2, \mathsf{s}_3 \\ \text{(o4')} & \mathsf{s}_0 \textbf{ precedes } \mathsf{s}_1 \\ \text{(o5')} & \mathsf{s}_1 \textbf{ precedes } \mathsf{s}_2 \\ \text{(o6')} & \mathsf{s}_2 \textbf{ precedes } \mathsf{s}_3 \\ \text{(o7')} & \neg \mathit{light\_on} \textbf{ at } \mathsf{s}_0 \\ \text{(o8')} & \mathit{light\_on} \textbf{ at } \mathsf{s}_1 \\ \text{(o9')} & \neg \mathit{light\_on} \textbf{ at } \mathsf{s}_2 \\ \text{(o10')} & \neg \mathit{light\_on} \textbf{ at } \mathsf{s}_3 \end{cases}$$

The main difference between the story in Example 1 and the story in Sys_1 is that: in Sys_1, the final observation of the light being not on is done *immediately* after turning on the lamp while in N_0 a gap between the action and the observation is allowed. The difference between (o3) and (o3') is due to that. □

We will now discuss when a system description needs a diagnosis and what a diagnosis is.

3.1 Our preferred notion – consistency based diagnosis

As discussed in the literature, there are many ways we can define diagnosis. In this subsection we describe our most preferred notion, which is based on the consistency-based approach.

Intuitively, we say a system needs a diagnosis, if the following assumptions are inconsistent with the observations (i) all components are initially fine, and (ii) no action that can break a component occurs. To define diagnosis, we assume that all components were initially operating normally, and we try to conjecture minimal action occurrences to account for the observations. Since the semantics of $\mathcal{L}$ minimizes action occurrences, all we need to do is to consider the various models of the narrative and extract our diagnosis from each.

Definition 2 (Necessity of Diagnosis) We say a system $Sys = (SD, COMPS, OBS)$ needs a diagnosis if the narrative $(SD \setminus SD_{ab}, OBS \cup OK_0)$ does not have a model.

Note that the notion of a system needing a diagnosis is not meant to capture the notion that there is some fault in the system. It is a much weaker notion. An alternative notion[4] is to require that there exists c, such that the narrative $(SD, OBS \cup OK_0) \not\models$ **currently** $\neg ab(c)$. According to this alternative definition, a system needs a diagnosis if we cannot conclusively entail that all components are operating normally in the current state. We now establish the notion of a diagnosis in terms of a diagnostic model.

Definition 3 (Diagnostic Model) Let $Sys = (SD, COMPS, OBS)$ be a system that needs a diagnosis. We say M is a diagnostic model of Sys if M is a model of the narrative $(SD, OBS \cup \mathcal{D}(\Delta, COMPS \setminus \Delta))$, where $\Delta \subseteq COMPS$, and there is no $\Delta' \subseteq COMPS$, $\Delta \subset \Delta'$ such that narrative $(SD, OBS \cup \mathcal{D}(\Delta', COMPS \setminus \Delta'))$ has a model.

We can now extract information about any particular situation from the diagnostic model. In particular,

[4] suggested by Michael Gelfond

Definition 4 (Diagnosis) Let $Sys = (SD, COMPS, OBS)$ be a system that needs a diagnosis. A diagnosis with respect to situation s is the set of components $\Delta \in COMPS$ such that there exists a diagnostic model M of Sys and $\Delta = \{c \mid ab(c)$ **at** s holds in $M\}$.

A diagnosis with respect to $\mathsf{s_c}$ is called a *current fluent diagnosis.*

A diagnosis Δ (wrt. a situation s) is *minimal* if there exists no diagnosis Δ' (wrt. s) such that $\Delta' \subset \Delta$.

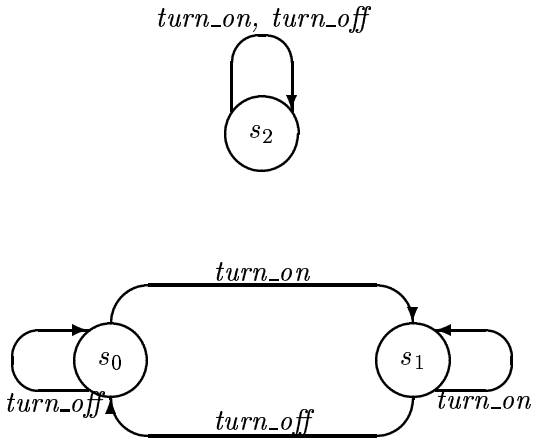
Example 4 (Continuation of Example 3) Consider the system $Sys_1 = (SD_1, \{bulb\}, OBS_1)$, from Example 3, with $SD_{ab} = \{break(bulb)$ **causes** $ab(bulb)\}$.

Let $N_1 = (SD_1 \backslash SD_{ab}, OBS_1 \cup OK_0)$. Due to the proposition "$\neg light\_on$ **if** $ab(bulb)$", $SD_1 \backslash SD_{ab}$ has only three distinct states: $s_0 = \emptyset$, $s_1 = \{light\_on\}$, and $s_2 = \{ab(bulb)\}$.

The transition function of $SD_1 \backslash SD_{ab}$ is given by

$$\Phi(turn\_on, s_0) = \{s_1\} \quad \Phi(turn\_on, s_1) = \{s_1\} \quad \Phi(turn\_on, s_2) = \{s_2\}$$
$$\Phi(turn\_off, s_0) = \{s_0\} \quad \Phi(turn\_off, s_1) = \{s_0\} \quad \Phi(turn\_off, s_2) = \{s_2\}$$

Φ can be represented graphically as follows.



We now prove that N_1 is inconsistent. Assume the contrary, N_1 has a model (Σ, Ψ). Because of the observations in $OBS_1 \cup OK_0$, we conclude that $\Psi([]) = s_0$. Let $\Sigma(\mathsf{s_2}) = \alpha$, where α is an action sequence. By the definition of a model of a narrative, we have that $\Sigma(\mathsf{s_3}) = \alpha \circ turn\_on$. As there is no action in $SD_1 \backslash SD_{ab}$ whose effect is $ab(bulb)$, we conclude that $ab(bulb) \notin \Psi(\alpha)$. This implies that $light\_on \in \Psi(\alpha \circ turn\_on)$, i.e., $light\_on$ must hold in $\mathsf{s_3}$. This contradicts the observation "$\neg light\_on$ **at** $\mathsf{s_3}$", i.e., N_1 is inconsistent.

Narrative N_1 is inconsistent, and hence, Sys_1 needs a diagnosis. We compute the diagnosis as follows.

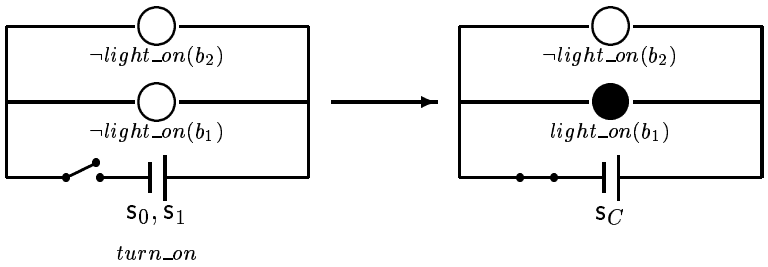
The narrative $(SD_1, OBS_1 \cup OK_0)$, however, has one model $M = (\Psi, \Sigma)$ where $\Psi([]) = s_0$ and

$\Sigma(\mathsf{s_0}) = \Sigma_1(\mathsf{s_1}) = []$,
$\Sigma(\mathsf{s_2}) = turn\_on$,
$\Sigma(\mathsf{s_3}) = turn\_on \circ turn\_off \circ break(bulb)$,
$\Sigma(\mathsf{s_4}) = \Sigma(\mathsf{s_C}) = turn\_on \circ turn\_off \circ break(bulb) \circ turn\_on$.

We can easily verify that $ab(bulb)$ **at** $\mathsf{s_C}$ holds in M. Hence a current diagnosis for Sys_1 is $\Delta = \{bulb\}$. Moreover, it is easy to check that Δ is also a minimal current diagnosis for Sys_1. □

Since $(SD_1, OBS_1 \cup OK_0)$ has a model, we can conclude that every component of Sys_1 is okay initially. In the next example, we consider a variation of an example from the literature in which some components are broken in the initial situation.

Example 5 (Adapted from [SD89]) Let us consider a variation of the three bulbs example from [SD89], represented by the following picture.



The system can be represented by $Sys_2 = (SD_2, \{b_1, b_2, battery\}, OBS_2)$ with $SD_2 = SD_2' \cup SD_{ab}$, $SD_{ab} = \{break(c)$ **causes** $ab(c) \mid c \in \{b_1, b_2, battery\}\}$,

$$SD_2' = \begin{cases} \text{(r1)} & turn\_on \textbf{ causes } light\_on(b_1) \textbf{ if } \neg ab(b_1), \neg ab(battery) \\ \text{(r2)} & turn\_on \textbf{ causes } light\_on(b_2) \textbf{ if } \neg ab(b_2), \neg ab(battery) \\ \text{(r3)} & \neg light\_on(b_1) \textbf{ if } ab(b_1) \vee ab(battery) \\ \text{(r4)} & \neg light\_on(b_2) \textbf{ if } ab(b_2) \vee ab(battery) \end{cases}$$

and

$$OBS_2 = \begin{cases} \text{(o1)} & \neg light\_on(b_1) \wedge \neg light\_on(b_2) \textbf{ at } s_0 \\ \text{(o2)} & light\_on(b_2) \wedge \neg light\_on(b_2) \textbf{ at } s_1 \\ \text{(o3)} & turn\_on \textbf{ between } s_0, s_1 \\ \text{(o4)} & s_0 \textbf{ precedes } s_1 \end{cases}$$

It is easy to check that no interpretation of $(SD_2 \setminus SD_{ab}, OBS_2 \cup OK_0)$ entails $\neg light\_on(b_2)$ **at** s_1. Thus, we need a diagnosis for Sys_2.

Moreover, no interpretation of $(SD_2, OBS_2 \cup OK_0)$ entails $\neg light\_on(b_2)$ **at** s_1.

The only diagnostic model of Sys_2 is the model of the narrative $(SD_2, OBS_2 \cup \mathcal{D}(\{b_1, battery\}, \{b_2\}))$ is $M = (\Psi, \Sigma)$ where

$\Psi([]) = \{ab(b_2)\}$, and

$\Sigma(s_0) = []$,
$\Sigma(s_1) = \Sigma(s_C) = turn\_on$.

Since $ab(b_2)$ **at** s_C holds in M, we have that $\Delta = \{b_2\}$ is a current diagnosis of Sys_2. □

3.2 Explanation vs diagnosis

Often the observations in a narrative can be *explained* by the sequence of actions (possibly exogenous) that have occurred. Unfortunately, this is not true in all cases because incomplete knowledge of the initial situation, and/or non-deterministic actions can lead to uncertainty in the outcome of a sequence of actions. We now formally clarify the difference between the notions of diagnosis and explanations.

- A system $(SD, COMPS, OBS)$ is said to be a *fully observed narrative* if $(SD \setminus SD_{ab}, OBS \cup OK_0)$ has a unique model M, and the action occurrences captured in M are explicitly mentioned in OBS.
- A system $(SD, COMPS, OBS)$ is said to need an explanation, if $(SD \setminus SD_{ab}, OBS \cup OK_0)$ is not a fully observed narrative.
- If a system $(SD, COMPS, OBS)$ is said to need an explanation, then if $(SD \setminus SD_{ab}, OBS \cup OK_0)$ has multiple models, then each model of $(SD \setminus SD_{ab}, OBS \cup OK_0)$ gives rise to an explanation of the system. On the other hand if $(SD \setminus SD_{ab}, OBS \cup OK_0)$ has no models then we say that the system needs a diagnosis.

Consider the narrative N_0 from Example 1. It follows from Example 2 that N_0 can be explained by the occurrence of a *turn_off* action just after the *turn_on* action in s_2 and before s_3. Thus, we do not need to worry about diagnosing this narrative. On the other hand there is no such explanation for the narrative $(SD_1 \setminus SD_{ab}, OBS_1 \cup OK_0)$, where no unobserved action is allowed between s_2 and s_3.

The definition of a diagnostic model in the previous section uses a *consistency* criterion and a *maximality* criterion to account for the observations. That is, the narrative (SD, Γ), where Γ comprises the sequence of action occurrences and initial situation (including OK_0) dictated by the diagnostic model, do not necessarily *entail* OBS. This may happen because actions can be non-deterministic. The following simple example illustrates this point.

Example 6 Consider the system $Sys = (SD, \{c_1, c_2\}, OBS)$ with

$$SD = \left\{ \begin{array}{lll} a_1 & \textbf{causes} & ab(c_1) \vee ab(c_2) \\ a_2 & \textbf{causes} & ab(c_1) \\ f & \textbf{if} & ab(c_1) \end{array} \right.$$

and

$$OBS = \left\{ \begin{array}{lll} \neg f & \textbf{at} & \mathsf{s}_0 \\ f & \textbf{at} & \mathsf{s}_1 \\ \mathsf{s}_0 & \textbf{precedes} & \mathsf{s}_1 \end{array} \right.$$

Obviously, $M = (\Psi_1, \Sigma_1)$ with $\Psi_1(\mathsf{s}_0) = \emptyset$, $\Psi_1(\mathsf{s}_1) = \{ab(c_1), f\}$ and $\Sigma_1(\mathsf{s}_0) = []$, $\Sigma_1(\mathsf{s}_1) = a_1$ is a diagnostic model for Sys. However, the narrative

$$(SD, OK(COMPS) \cup \{\neg f \ \textbf{at} \ \mathsf{s}_0,\ a_1 \ \textbf{occurs\_at} \ \mathsf{s}_0,\ \mathsf{s}_0 \ \textbf{precedes} \ \mathsf{s}_1\})$$

does not entail "f **at** s_1" because it has a model (Ψ_2, Σ_2) with $\Psi_2(\mathsf{s}_0) = \emptyset$, $\Psi_2(\mathsf{s}_1) = \{ab(c_2)\}$ and $\Sigma_2(\mathsf{s}_0) = []$, $\Sigma_2(\mathsf{s}_1) = a_1$ in which "f **at** s_1" does not hold. □

We now define the notion of an explanatory diagnostic model, which has the stronger criterion that (SD, Γ) must entail the observations.

Definition 5 (Explanatory Diagnostic Model) Suppose M is a model of the narrative $(SD, OBS \cup \mathcal{D}(\Delta, COMPS \setminus \Delta))$ and is a diagnostic model of $(SD, COMPS, OBS)$, where

- $actions(M)$ is the set of occurrence facts and precedence facts of the forms (5), (6), and (7), (i.e., facts of the forms α **between** $\mathsf{s}_1, \mathsf{s}_2$, α **occurs_at** s_1, s_1 **precedes** s_2) that are true in M; and

- $initial(M)$ is the set of fluent facts of the form f **at** s_0 that are true in M, including $\mathcal{D}(\Delta, COMPS \setminus \Delta)$.

Then M is an *explanatory diagnostic model* iff

$$(SD,\ actions(M) \cup initial(M)) \models OBS.$$

Following in this spirit, it is straightforward to define the notion of an explanatory diagnosis as the set of action occurrences of some explanatory diagnostic model, i.e., if M is an explanatory diagnostic model for Sys, then $actions(M)$ is an explanatory diagnosis for Sys.

The following proposition gives conditions on a system when our notion of explanatory diagnosis is equivalent to our notion of consistency-based diagnosis.

Proposition 1 Let $Sys = (SD, COMPS, OBS)$ be a system that needs a diagnosis. If all actions in SD are deterministic and Sys has complete information about the initial situation then M is a diagnostic model iff M is an explanatory diagnostic model.

Proof. By definition each explanatory diagnostic model is a diagnostic model. Thus, to prove the proposition, we need to show that if $M = (\Psi, \Sigma)$ is a diagnostic model of Sys that satisfies the conditions in the proposition, then M is an explanatory diagnostic model.

Assume that M is a model of the narrative $(SD, OBS \cup \mathcal{D}(\Delta, COMPS \setminus \Delta))$ for some $\Delta \subseteq COMPS$ and is a diagnostic model for Sys. We need to prove that

$$(SD, actions(M) \cup initial(M)) \models OBS$$

where $actions(M)$ is the set of occurrence facts and precedence facts which hold in M and $initial(M)$ is the set of fluent facts of the form f **at** s_0 that are true in M, including $\mathcal{D}(\Delta, COMPS \setminus \Delta)$.

Let $Q = (SD, actions(M) \cup initial(M))$. We need to show that OBS is true in every model (Ψ', Σ') of Q. We prove it by showing the following: (i) M is a model of Q, (ii) OBS is true in M, and (iii) M is the unique model of Q.

Since M is a diagnostic model of Sys, we know that Ψ is a causal model of Q, and, $initial(M)$ and $actions(M)$ are true in M. Furthermore, $\Sigma(\mathsf{s}_C)$ **occurs_at** s_0 is true in M, and hence, it a fact in Q. Thus, there exists no proper subsequence of $\Sigma(\mathsf{s}_C)$ that satisfies the facts of Q. Thus, M is a model of Q and OBS is true in M. (1)

Assume that $M' = (\Psi', \Sigma')$ is any arbitrary model of Q. Since we have complete information about the initial situation, we have that $\Psi([]) = \Psi'([])$. This, together with the fact that actions in SD are deterministic imply that $\Psi(\alpha) = \Psi'(\alpha)$ for every sequence of actions α. To show that M is the unique model of Q we now need to show that $\Sigma(\mathsf{s}_i) = \Sigma'(\mathsf{s}_i)$ for every $i = 0, \ldots, C$. This follows immediately from the observation that $\Sigma(\mathsf{s}_i)$ **between** $\mathsf{s}_0, \mathsf{s}_i$ is true in M for every $i = 0, \ldots, C$, and hence, is an occurrence fact in Q. Since M' is a model of Q, we have that $\Sigma'(\mathsf{s}_i) = \Sigma(\mathsf{s}_0) \circ \Sigma(\mathsf{s}_i) = \Sigma(\mathsf{s}_i)$ for every $i = 0, \ldots, C$. Hence, M is a unique model of Q. (2)

It follows from (1) and (2) that $Q \models OBS$. The proposition is proved. □

3.3 Relationship between dynamic diagnosis and diagnosis from first principles

In this subsection we explore the relationship between our framework of diagnosis and the notions of diagnosis from first principles as described in [dKMR92, Rei87]. We start with recalling the definitions there.

There, a system description is a triple $Sys = (SD, COMPS, OBS)$ where $COMPS = \{c_1, \ldots, c_n\}$ is a set of components, OBS is a set of observations, and SD is the behavioral specification of the system described as a logical theory in a language whose signature contains an unary predicate ab. In addition, we will require that $SD \cup \{\neg ab(c_1), \ldots, \neg ab(c_n)\}$ be consistent. (We were unable to find a reference in which this assumption is made explicitly. It seems that the assumption should follow from the intuition that the behavioral specification of the system allows the possibility that none of its components are faulty.)

Sys is said to be inconsistent wrt. OBS, if $SD \cup \{\neg ab(c_1), \ldots, \neg ab(c_n)\} \cup OBS$ is unsatisfiable. In this case, we need to diagnose the system.

A *diagnosis* (or *minimal diagnosis*) for such a system is defined as a *minimal subset* Δ of $COMPS$ such that $SD \cup OBS \cup \{ab(c) \mid c \in \Delta\} \cup \{\neg ab(c) \mid c \in COMPS \setminus \Delta\}$ is satisfiable.

Several alternatives of minimal diagnosis such as *kernel diagnosis, partial diagnosis, prime diagnosis,* or *abductive diagnosis* [dKMR92] have been discussed in the model-based diagnosis literature. They differ from the minimal diagnosis by the formula over the set of abnormal literals which is required to be consistent. As such, we will only discuss the relationship between our notion of consistency-based diagnosis and the notion of minimal diagnosis in this paper. We believe that dynamic counterparts to the variations of minimal diagnosis can be defined correspondingly, with a little modification to our notion of consistency-based diagnosis.

3.3.1 Similarities and differences: an intuitive analysis

We believe that most (or perhaps all) systems are inherently dynamic and actions (or events) occur when the system transitions from one state to another. So, one way to view the first principle formulation of diagnosis is to view SD as the specification of *the various states the system may be in (the system states)*, and OBS as observations about one such state. On the other hand, in our formulation using action theories, *the set of states that the system may be in (the system states)* is given by the set of states that can be reached by a valid sequence of actions in our domain, starting from the set of initial states that must satisfy the static causal laws and have all components as not abnormal.

Our approach is *more elaboration tolerant* in the sense that it allows us to easily change the specification of the system states. This is extremely useful. Let us consider the following case. Suppose while doing diagnosis we decide that a certain action is extremely unlikely, and we would like to find a diagnosis assuming that it could not have occurred. It is extremely easy to update our system description - by just removing that action from the language – to accommodate this. On the other hand such elaboration tolerance is not a requirement in the specification of SD, in the first principle formulation of diagnosis.

Another advantage of our approach is that our OBS allows observations to be about multiple situations. This is not a requirement in the first principle formulation and does not seem to be compatible to the definition of diagnosis there as then it is not clear about which situation we are diagnosing. Assuming then that in the first principle formulation OBS is about a single situation, accounting for action occurrences and observations about different situations would then need

progression or regression of the various observations – at possibly different situations – to a single situation. This is fairly cumbersome compared to allowing observations about different situations and adding them to OBS directly, as done in our formulation.

3.3.2 Similarities: some formal results

In this subsection we show how the notion of diagnosis from first principles can be captured as a special case in our formulation. Given a system description $Sys = (SD, COMPS, OBS)$ in the first principle notation, we construct a system description $Sys' = (SD', COMPS, OBS')$ in our notation such that there is correspondence between diagnosis in Sys and diagnosis in Sys'. (We will refer to Sys' as the dynamic version of Sys.)

Sys' is constructed as follows:

- SD' consists of:
  - static causal laws: $\tilde{SD} = \{\varphi \textbf{ if } true \mid \varphi \in SD\}$; and
  - dynamic causal laws about actions that can make any fluent formula to hold: $\{make(\varphi) \textbf{ causes } \varphi \mid \varphi \text{ is fluent formula in } Sys\}$.
- OBS' consists of

  $\{\varphi \textbf{ at } \mathsf{s}_1 \mid \varphi \in OBS\} \cup \{\mathsf{s}_0 \textbf{ precedes } \mathsf{s}_1\}$.

The following proposition relate diagnosis in Sys with diagnosis in Sys'.

Proposition 2 Let $Sys = (SD, COMPS, OBS)$ be a system description and Sys' be the dynamic version of Sys. Then, Δ is a minimal current fluent diagnosis for Sys' iff Δ is a diagnosis for Sys.

Proof.

($\Longrightarrow$)
Let Δ be a minimal current fluent diagnosis for Sys'. By Lemma 3 (Appendix), we have that $SD \cup OBS \cup \{ab(c) \mid c \in \Delta\} \cup \{\neg ab(c) \mid c \in COMPS \setminus \Delta\}$ is satisfiable. We need to show that Δ is a diagnosis for Sys. Assume the contrary, Δ is not a diagnosis for Sys. This implies that there exists a diagnosis Δ' for Sys such that $\Delta' \subset \Delta$. By Lemma 2 (Appendix), Δ' is a current fluent diagnosis for Sys'. This contradicts the fact that Δ is a minimal current fluent diagnosis for Sys'. Thus, our assumption is incorrect, i.e., we have proved that Δ is a diagnosis for Sys. (1)

($\Longleftarrow$)
Let Δ be a diagnosis for Sys. By Lemma 2, we have that Δ is a current fluent diagnosis for Sys'. We need to show that Δ is a minimal current fluent diagnosis for Sys'. Assume the contrary, there exists a current fluent diagnosis Δ' for Sys' such that $\Delta' \subset \Delta$. By Lemma 3, $SD \cup OBS \cup \{ab(c) \mid c \in \Delta'\} \cup \{\neg ab(c) \mid c \in COMPS \setminus \Delta'\}$ is satisfiable, which implies that Δ is not a diagnosis for Sys. This contradicts the fact that Δ is a diagnosis for Sys. Thus, our assumption is incorrect, i.e., we have proved that Δ is a minimal current fluent diagnosis for Sys. (2)

The conclusion of the theorem follows from (1) and (2). □

3.4 Multiple fault modes

So far we have been assuming that a component can be either abnormal or normal. Often a component can have various abnormal states, and it may be important to distinguish which particular abnormal state it is in. For example, in [WN96] an abnormal valve can be in several states: it could stuck at the ON position, it could be stuck at the OFF position, or it could behave completely randomly.

A simple way to accommodate this without changing much in our formulation is to have another 2-ary ab predicate, where $ab(c, m)$ means that the component c is in the abnormal mode m, and use such predicates in the description. In addition we will have the following two causal rules – where $\oplus$ is the ex-or symbol – for each component:

$$ab(c) \Rightarrow ab(c, m_1) \oplus \ldots \oplus ab(c, m_{n_c}) \tag{9}$$

$$\neg ab(c) \Rightarrow \neg ab(c, m_1) \wedge \ldots \wedge \neg ab(c, m_{n_c}) \tag{10}$$

The rest of our formulation remains unchanged.

4 Diagnostic and repair planning

The diagnostic process discussed in the previous section will generate a set of candidate diagnoses, however diagnosis is only the first step in dealing with an errant system. In most cases we will attempt to discriminate these diagnoses with the objective of identifying a unique diagnosis and/or reducing our space of candidate diagnoses to a point where a repair plan can be conceived. We are operating under the assumption that we cannot directly observe the state of abnormality of the various components of the system. Nevertheless, we can make other observations about the system, add them to OBS, and then refine our diagnoses.

In general, the fluents in the system are of two kinds: *observable* and *unobservable*. A simple *generic observation*[5] leads the agent to know the value of the observable fluents. By knowing the relationship between the observable and unobservable fluents, and the values of the observable fluents, we can sometimes deduce the values of unobservable fluents. We can also use direct sensing actions to sometimes determine their value. Given a set of candidate diagnoses, we can execute a plan – perhaps including some sensing actions and conditional branches – and make the generic observations to obtain additional information that will help reduce the space of possible diagnoses. Such plans are distinguished in that they can have knowledge goals in addition to goals relating to the state of the world. We refer to plans that attempt to reduce our space of diagnoses as *diagnostic plans*. A diagnostic plan that includes some repair is called a *repair plan*.

The main goal of this section is to formally define a diagnostic plan. In the process, we must extend our action language with sensing actions and also with the notions of *observable* and *unobservable fluents*. But first we will illustrate the role of diagnostic planning through a simple example.

Example 7 Let Sys_3 be the system $(SD_3 \cup SD_{ab}, \{bulb, switch\}, OBS_3)$ where

[5] Here we distinguish between generic observations and sensing actions. We assume that the agent is constantly performing 'generic observations' and thus knows the truth value of the observable fluents at all times. In contrast, sensing actions require the agent's effort.

$$SD_3 = \left\{ \begin{array}{ll} \text{(r1)} & \mathit{turn\_on} \textbf{ causes } \mathit{light\_on} \\ & \quad \textbf{if } \neg \mathit{ab}(\mathit{bulb}), \neg \mathit{ab}(\mathit{switch}), \mathit{connected}(\mathit{bulb}, \mathit{switch}) \\ \text{(r2)} & \mathit{turn\_off} \textbf{ causes } \neg \mathit{light\_on} \\ & \quad \textbf{if } \mathit{connected}(\mathit{bulb}, \mathit{switch}) \\ \text{(r3)} & \mathit{disconnect}(\mathit{bulb}, \mathit{switch}) \textbf{ causes } \neg \mathit{connected}(\mathit{bulb}, \mathit{switch}) \\ & \quad \textbf{if } \mathit{connected}(\mathit{bulb}, \mathit{switch}) \\ \text{(r4)} & \mathit{connect}(\mathit{bulb}, \mathit{test\_device}) \textbf{ causes } \mathit{connected}(\mathit{bulb}, \mathit{test\_device}) \\ & \quad \textbf{if } \neg \mathit{connected}(\mathit{bulb}, \mathit{switch}) \\ \text{(r5)} & \mathit{light\_on} \textbf{ if } \mathit{connected}(\mathit{bulb}, \mathit{test\_device}), \neg \mathit{ab}(\mathit{bulb}) \end{array} \right.$$

$$OBS_3 = \left\{ \begin{array}{ll} \text{(o1)} & \mathit{turn\_on} \textbf{ occurs\_at } \mathsf{s}_0 \\ \text{(o2)} & \mathit{turn\_off} \textbf{ occurs\_at } \mathsf{s}_1 \\ \text{(o3)} & \mathit{turn\_on} \textbf{ between } \mathsf{s}_2, \mathsf{s}_3 \\ \text{(o4)} & \mathsf{s}_0 \textbf{ precedes } \mathsf{s}_1 \\ \text{(o5)} & \mathsf{s}_1 \textbf{ precedes } \mathsf{s}_2 \\ \text{(o6)} & \mathsf{s}_2 \textbf{ precedes } \mathsf{s}_3 \\ \text{(o7)} & \neg \mathit{light\_on} \textbf{ at } \mathsf{s}_0 \\ \text{(o8)} & \mathit{connected}(\mathit{bulb}, \mathit{switch}) \textbf{ at } \mathsf{s}_0 \\ \text{(o9)} & \neg \mathit{connected}(\mathit{bulb}, \mathit{test\_device}) \textbf{ at } \mathsf{s}_0 \\ \text{(o10)} & \mathit{light\_on} \textbf{ at } \mathsf{s}_1 \\ \text{(o11)} & \neg \mathit{light\_on} \textbf{ at } \mathsf{s}_2 \\ \text{(o12)} & \neg \mathit{light\_on} \textbf{ at } \mathsf{s}_3 \end{array} \right.$$

and $SD_{ab} = \{\mathit{break}(\mathit{bulb}) \textbf{ causes } \mathit{ab}(\mathit{bulb}),\ \mathit{break}(\mathit{switch}) \textbf{ causes } \mathit{ab}(\mathit{switch})\}$. Here, the only observable fluent is *light_on*.

It is easy to see that there are two current fluent diagnoses for Sys_3: $\Delta_1 = \{\mathit{bulb}\}$ and $\Delta_2 = \{\mathit{switch}\}$ which correspond to the models (Ψ, Σ_1) and (Ψ, Σ_2) of $(SD_3 \cup SD_{ab}, OBS_3 \cup OK_0)$ where

$\Psi([]) = \{\mathit{connected}(\mathit{bulb}, \mathit{switch})\}$, and

$\Sigma_1(\mathsf{s}_0) = []$,
$\Sigma_1(\mathsf{s}_1) = \mathit{turn\_on}$,
$\Sigma_1(\mathsf{s}_2) = \mathit{turn\_on} \circ \mathit{turn\_off} \circ \mathit{break}(\mathit{bulb})$,
$\Sigma_1(\mathsf{s}_3) = \Sigma_1(\mathsf{s}_C) = \mathit{turn\_on} \circ \mathit{turn\_off} \circ \mathit{break}(\mathit{bulb}) \circ \mathit{turn\_on}$, and

$\Sigma_2(\mathsf{s}_0) = []$,
$\Sigma_2(\mathsf{s}_1) = \mathit{turn\_on}$,
$\Sigma_2(\mathsf{s}_2) = \mathit{turn\_on} \circ \mathit{turn\_off} \circ \mathit{break}(\mathit{switch})$,
$\Sigma_2(\mathsf{s}_3) = \Sigma_2(\mathsf{s}_C) = \mathit{turn\_on} \circ \mathit{turn\_off} \circ \mathit{break}(\mathit{switch}) \circ \mathit{turn\_on}$. □

In our example, there are two components: bulb and switch. When the light does not turn on after turning the switch on, the diagnosis is that either the bulb is abnormal or the switch is broken (i.e., abnormal). To get to a unique diagnosis we can take the bulb out and connect it to a testing device and observe if the bulb works with the testing device. If it does, it means that the bulb is fine and hence the switch must be broken. (Now to fix the system we have to fix the switch and

also remember to put the bulb back on.) If the bulb does not work then the bulb is broken. If we us the notion of minimal diagnosis we will conclude that the switch is fine.

The above example shows that we could identify the defective components of the system by executing a simple plan (a sequence of actions) and observing its result. In the above example, the plan is *turn_off* $\circ$ *get_bulb* $\circ$ *put_bulb* and the observation is *light_on*.

4.1 Adding sensing and observables to $\mathcal{L}$

In this section we augment the domain description language $\mathcal{L}_D$, observation language $\mathcal{L}_O$, and the query language $\mathcal{L}_Q$ so as to incorporate sensing actions and also the notion of observable and unobservable fluents.

4.1.1 $\mathcal{L}_{DS}$: The Domain Description Language

Like $\mathcal{L}_D$, the alphabets of $\mathcal{L}_{DS}$ consists of two nonempty, disjoint sets of symbols $\mathbf{F}$ (fluents) and $\mathbf{A}$ (actions). We distinguish a set of *observable fluents* $\mathbf{F_O}$ from the set of fluents $\mathbf{F}$. In addition to propositions of the forms (1)-(3), $\mathcal{L}_{DS}$ allows a new type of proposition, called as *knowledge producing laws*, of the following form:

$$\mathsf{a} \textbf{ determines } f \tag{11}$$

where a is an action and f is a fluent. A law of this form tells us that after a is executed, the value of the fluent f will be known. An action occurring in a knowledge producing law is called a *sensing action*. In the following, we will assume that sensing actions do not occur in dynamic causal laws.

As in $\mathcal{L}_D$, an $\mathcal{L}_{DS}$ domain description D is a set of propositions of the forms (1)-(3) or (11).

We now describe how the formulation of sensing actions in [BS98] (which was inspired by the formulations in [Moo85, SL93]) is incorporated in $\mathcal{L}_{DS}$.

Since we are now dealing with incomplete information, we now need to distinguish between a state of the world and the state of the agent's knowledge about the world. The later will be referred to as a *knowledge state* (or *k-state*) and will be represented by a set of states. We define a *combine state* (or *c-state*) as a pair of the form $\langle s, \mathcal{S}\rangle$, where s is a state and $\mathcal{S}$ is a set of states (or a k-state). Intuitively, in a c-state $\langle s, \mathcal{S}\rangle$, s represents the real state of the world whereas $\mathcal{S}$ is the set of states an agent thinks it may be in.

Two states s and s' are said to *agree on a literal* f, denoted by $s \sim_f s'$, if $f \in s$ iff $f \in s'$. If s and s' agree on a set of literals Z, we write $s \sim_Z s'$. A fluent formula φ is known to be true (resp. false) in a c-state $\langle s, \mathcal{S}\rangle$ if φ (resp. $\neg\varphi$) holds in every state $s' \in \mathcal{S}$; and φ is *known* in $\langle s, \mathcal{S}\rangle$, if φ is known to be true or known to be false in $\langle s, \mathcal{S}\rangle$.

For a sensing action a, $\Phi(a, s) = \{s\}$ if there exists no executability condition **impossible** a **if** φ in D such that φ holds in s. Otherwise $\Phi(a, s) = \emptyset$. In general, an action a *is said to be executable* in a c-state $\langle s, \mathcal{S}\rangle$ if $\Phi(a, s) \neq \emptyset$.

Given a domain description D in $\mathcal{L}_{DS}$, by $\mathcal{K}$ we denote the set of c-states of D, i.e., $\mathcal{K} = 2^{\mathbf{F}} \times 2^{2^{\mathbf{F}}}$. We now extend the transition function Φ to also map pairs of actions and c-states into sets of c-states. (Note that we are overloading Φ here, and for an action a, a state s, and a set of states $\mathcal{S}$, by $\Phi(a, s)$ and $\Phi(a, \mathcal{S})$ we denote $\{s' \mid s' = Cn(e_a(s) \cup (s \cap s') \cup R)\}$ and $\{s' \mid \exists s^+ \in \mathcal{S} \text{ s.t. } s' \in \Phi(a, s^+)\}$ respectively.)

The mapping of action and c-states by Φ is now defined as follows:

1. for any c-state $\langle s, \mathcal{S}\rangle$ and non-sensing action a,

   $\Phi(a, \langle s, \mathcal{S}\rangle) = \{\langle s', \mathcal{S}'\rangle \mid s' \in \Phi(a, s)$, and $\mathcal{S}'$ is the set of states in $\Phi(a, \mathcal{S})$ that agree with s' on the literals from $\mathbf{F_O}\}$.

   (Note that if a is not executable in $\langle s, \mathcal{S}\rangle$ then $\Phi(a, \langle s, \mathcal{S}\rangle) = \emptyset$.)

2. for any c-state $\langle s, \mathcal{S}\rangle$ and sensing action a whose knowledge producing laws are
   a **determines** f_1 ... a **determines** f_m

   (a) if a is executable in $\langle s, \mathcal{S}\rangle$, $\Phi(a, \langle s, \mathcal{S}\rangle) = \{\langle s, \{s' \mid s' \in \mathcal{S}$ such that s and s' agree on the literals from $\mathbf{F_O} \cup \{f_1, \ldots, f_m\}\}\rangle\}$;

   (b) otherwise, $\Phi(a, \langle s, \mathcal{S}\rangle) = \emptyset$.

Proposition 3 For every $\mathcal{L}_{DS}$ domain description D the transition function Φ of D is unique. □

4.1.2 $\mathcal{L}_{OS}$: The observation language

The observation language $\mathcal{L}_{OS}$ is same as the observation language $\mathcal{L}_O$. Note that we do allow action occurrences about sensing actions (i.e., of propositions of the form (5) and (6)). But since, after executing sensing actions the agent will know the value of fluents sensed by those sensing actions, we require that fluent facts – of the form (4) – to that effect are part of the observation. Despite this requirement, the reason we allow action occurrences about sensing actions is because, the occurrences together with the executability conditions may give us additional information about the trajectory.

The characterization of narratives that include sensing actions in the domain description and the observation part, remain essentially the same as before, with the only addition that we now have defined what $\Phi(a, s)$ means for a sensing action a. The important thing to note is that, in characterizing a narrative, we do not use the mapping given by Φ with respect to action, and c-state pairs. This will be later used only to reason about plans.

4.1.3 $\mathcal{L}_{QS}$: The query language

In the presence of incomplete information and knowledge producing actions, there may not exist simple plans consisting of sequence of actions and we may need to extend the notion of a plan to allow conditional statements. We refer to such plans as a conditional plan.

In the literature [BS98, SL93], queries in action domains with sensing actions are of the forms:

$$\textbf{knows}\ \varphi\ \textbf{after}\ \ P\ \ \textbf{at}\ \ \mathsf{s} \tag{12}$$

$$\textbf{whether}\ \varphi\ \textbf{after}\ \ P\ \ \textbf{at}\ \ \mathsf{s} \tag{13}$$

where φ is a fluent formula and P is a conditional plan as formally defined below. In this paper, besides (12) and (13) we are also interested in query of the form

$$\varphi\ \textbf{during}\ P\ \ \textbf{at}\ \ \mathsf{s} \tag{14}$$

where, as in other types of queries, φ is a fluent formula and P is a conditional plan.

Definition 6 (Conditional Plan) 1. An empty sequence of action, denoted by [ ], is a conditional plan.

2. If a is an action then a is a conditional plan.

3. If $P_1, \ldots, P_n$ are conditional plans and φ_j's are conjunction of fluent literals (which are mutually exclusive but not necessarily exhaustive) then the following is a conditional plan. (We refer to such a plan to as a *case plan*).

   Case
   $\varphi_1 \rightarrow P_1$
   $\ldots$
   $\varphi_n \rightarrow P_n$
   Endcase

4. If P_1 and P_2 are conditional plans then $P_1; P_2$ is a conditional plan.

5. Nothing else is a conditional plan. □

In order to define the entailment of queries in $\mathcal{L}_{QS}$ that include conditional plans, from narratives, we need to define an *extended transition function* $\hat{\Phi}$, which maps a pair of a conditional plan and a c-state into a set of c-states. Intuitively, if $\sigma' \in \hat{\Phi}(P, \sigma)$ then it means that execution the plan in the c-state σ may take us to the c-state σ'. Before defining $\hat{\Phi}$, we first define the possible trajectories when P is executed in σ.

Definition 7 Let P be a conditional plan and σ be a c-state. We say a sequence of c-states $\sigma_1, \ldots, \sigma_n$ is a trajectory of P wrt. σ if:

1. $P = []$, and $n = 1$, and $\sigma_1 = \sigma$.

2. $P = [a]$, and $n = 2$, and $\sigma_1 = \sigma$ and $\sigma_2 \in \Phi(a, \sigma)$.

3. $P =$ Case
   $\varphi_1 \rightarrow P_1$
   $\ldots$
   $\varphi_n \rightarrow P_n$
   Endcase,

   and there exists an i such that φ_i is known to be true in σ
   and $\sigma_1, \ldots, \sigma_n$ is a trajectory of P_i wrt. σ.

4. $P = P_1; P_2$, and $\sigma_1 = \sigma$, and $\sigma_1, \ldots, \sigma_k$ is a trajectory of P_1 wrt. σ, and $\sigma_{k+1}, \ldots, \sigma_n$ is a trajectory of P_2 wrt. σ_{k+1}.

σ_n is referred to as the resulting c-state of P wrt. σ.

□

Let P be a conditional plan and $\sigma = \langle s, \mathcal{S} \rangle$ be a c-state, $\hat{\Phi}(P, \sigma)$ is now defined as follows:

Definition 8 1. $\hat{\Phi}([], \sigma) = \{\sigma\}$;

2. For an action a, $\hat{\Phi}(a,\sigma) = \Phi(a,\sigma)$;

3. For $P =$ Case
   $\varphi_1 \rightarrow P_1$
   $\cdots$
   $\varphi_n \rightarrow P_n$
   Endcase,

$$\hat{\Phi}(P,\sigma) = \begin{cases} \hat{\Phi}(P_i,\sigma) \text{ if } \varphi_i \text{ is known to be true in } \sigma \\ \\ \emptyset \text{ if there exists no } i \text{ s.t. } \varphi_i \text{ is known to be true in } \sigma \end{cases}$$

4. For $P = P_1; P_2$, where P_1 is a conditional plan and P_2 is a conditional plan,

   - if $\hat{\Phi}(P_1,\sigma) \neq \emptyset$, and for every $\sigma' \in \hat{\Phi}(P_1,\sigma)$, $\hat{\Phi}(P_2,\sigma') \neq \emptyset$, then $\hat{\Phi}(P,\sigma) = \bigcup_{\sigma' \in \hat{\Phi}(P_1,\sigma)} \hat{\Phi}(P_2,\sigma')$; and
   - $\hat{\Phi}(P,\sigma) = \emptyset$ otherwise.

□

It should be noted that $\hat{\Phi}(P,\sigma)$ is not equal to the set of resulting c-states of P wrt. σ. This is because some branches of P may lead to unexecutable actions and hence $\hat{\Phi}(P,\sigma)$ will be empty while there may be several trajectories corresponding to other branches.

Our next goal is to define entailment of queries w.r.t narratives. This entailment basically defines correctness of conditional plans and gives insight into what planning with incompleteness and sensing w.r.t a narrative means. Intuitively, since the narrative may not be complete (or does not have sufficient observations) to arrive at a unique model, multiple models may tell us that a situation s may correspond to many different states, only one of which corresponds to s in reality. Thus we have a set of c-states from which we need to verify the correctness of a conditional plan with respect to a goal. More formally,

Definition 9 (Possible State wrt. a Situation) Let $N = (D,\Gamma)$ be a narrative. We say s is a possible state corresponding to situation s, if there exists a model (Ψ,Σ) of N such that $\Psi(\Sigma(\mathsf{s})) = s$. We say $\sigma = \langle s,\mathcal{S}\rangle$ is a c-state corresponding to situation s, if s is a possible state corresponding to situation s and $\mathcal{S}$ is the set of all possible states corresponding to situation s.

Definition 10 (Entailement) Let $N = (D,\Gamma)$ be a narrative. A query q of $\mathcal{L}_{QS}$ is said to be entailed by (D,Γ), denoted by $(D,\Gamma) \models q$, if for every c-state $\langle s,\mathcal{S}\rangle$ corresponding to s, $\hat{\Phi}(P,\langle s,\mathcal{S}\rangle) \neq \emptyset$ and

- if $q =$ **knows** φ **after** P **at** s then φ is known to be true in every c-state belonging to $\hat{\Phi}(P,\langle s,\mathcal{S}\rangle)$; or
- if $q =$ **whether** φ **after** P **at** s, then for every c-state $\langle s',\mathcal{S}'\rangle$ belonging to $\hat{\Phi}(P,\langle s,\mathcal{S}\rangle)$ and for every state $s'' \in \mathcal{S}'$, $s' \sim_\varphi s''$; or
- if $q = \varphi$ **during** P **at** s then for all trajectories of the form $\langle s_1,\mathcal{S}_1\rangle,\ldots,\langle s_n,\mathcal{S}_n\rangle$ of P wrt. $\langle s,\mathcal{S}\rangle$, $s_i \sim_\varphi s_j$ for $1 \leq i \neq j \leq n$. □

Definition 11 (Temporal Knowledge Plan) Let $N = (D, \Gamma)$ be a narrative, L_M, L_W, and L_K be conjunctions of literals. A plan P is a *temporal knowledge plan* wrt. $(L_M,\ L_W,\ L_K)$ if

- $(D, \Gamma) \models \bigwedge_{l \in L_M} l$ **during** P **at** s_C;
- $(D, \Gamma) \models$ **knows** $\bigwedge_{l \in L_W} l$ **after** P **at** s_C; and
- $(D, \Gamma) \models$ **whether** $\bigwedge_{l \in L_K} l$ **after** P **at** s_C.

4.2 Diagnostic and repair plans

We are now ready to define what a diagnostic plan is. Intuitively, it is a conditional plan, possibly with sensing actions which when executed in the current situation gives us enough information to reach a unique diagnosis. In addition, we may have certain restrictions, such as:

- Certain literals are not allowed to change their values during the execution of the plan. We will refer to such literals as *protected literals.*
- Certain literals are allowed to change during the execution of the plan, but we require that at the end of the execution of the plan, their value be the same as it was before the plan was executed. We refer to such literals as *restored literals.*
- Certain literals are allowed to change during the execution of the plan, but we require that at the end of the plan, their value be either the same as it was before the plan was executed or be $false$. Such literals will be referred to as *fixable literals.* (This accommodates repair, where ab fluents can be made $\neg ab$.)

The intuition behind the above restrictions are as follows. Sometimes we do not want to affect certain aspects of a system while diagnosing. These aspects may be too dangerous or too valuable to tinker with, even if our intention is to revert them to their original state after changing their state. Also, changing their state may defeat the whole purpose. For example, to stabilize the leaning tower of Pisa we may not break it and rebuild it. Thus our option is to do unobtrusive sensing (perhaps through X-rays) to diagnose. Fluent literals corresponding to such aspects are labeled as *protected literals.* For certain other aspects of the system, we are allowed to tinker with them while diagnosing but we need to bring them back to their original stage. Examples of such aspects include, opening a flashlight to differentiate between if the bulb is bad, or if the batteries are down, or if the connection is bad. But while opening the flashlight, even if the connection was initially fine, we change the state of the connection. Thus we are required to put the flashlight back so that the state of the connection reverts back to its original state. Another similar example is the disassembling of a car engine to diagnose it. We need to put it back. Fluent literals corresponding to such aspects are labeled as *restored literals* . If we allow integration of repair with diagnostic planning, then while diagnosis or execution of the diagnostic plan we may allow that certain abnormal components are fixed but not vice-versa. Such literals are labeled as *fixable* literals.

We now define diagnostic plans as follows.

Definition 12 (Diagnostic Plan) Given a system $Sys = (SD, COMPS, OBS)$. A temporal knowledge plan of (SD, OBS) wrt. (L_M, L_W, L_K) is a *diagnostic plan* of Sys wrt. (L_M, L_W, L_K) if

- $\{ab(c) \mid c \in COMPS\} \cup \{\neg ab(c) \mid c \in COMPS\} \subseteq L_M$ and

- $L_K \subseteq \{ab(c) \mid c \in COMPS\}$.

In particular,

- if $L_K = \{ab(c) \mid c \in COMPS\}$ and $L_W = \emptyset$, we say that P is a *purely diagnostic plan* of Sys.
- if $L_K = \{ab(c)\}$ for some $c \in COMPS$, we say that P is a *discriminating diagnostic plan* of Sys for c.

□

We illustrate this definition by considering diagnostic plans for the systems Sys_1, Sys_2, and Sys_3 in the next example.

Example 8 Since Sys_1 and Sys_2 have only one diagnosis, the empty plan (i.e. []) is a purely diagnostic plan for Sys_1 and Sys_2. We will prove that the empty plan is not a purely diagnostic plan.

Recall that the only observable fluent in Sys_3 is *light_on*.

It follows from Example 7 that there are two possible current states:

$s_1 = \{connected(bulb, switch), ab(bulb)\}$ and $s_2 = \{connected(bulb, switch), ab(switch)\}$.

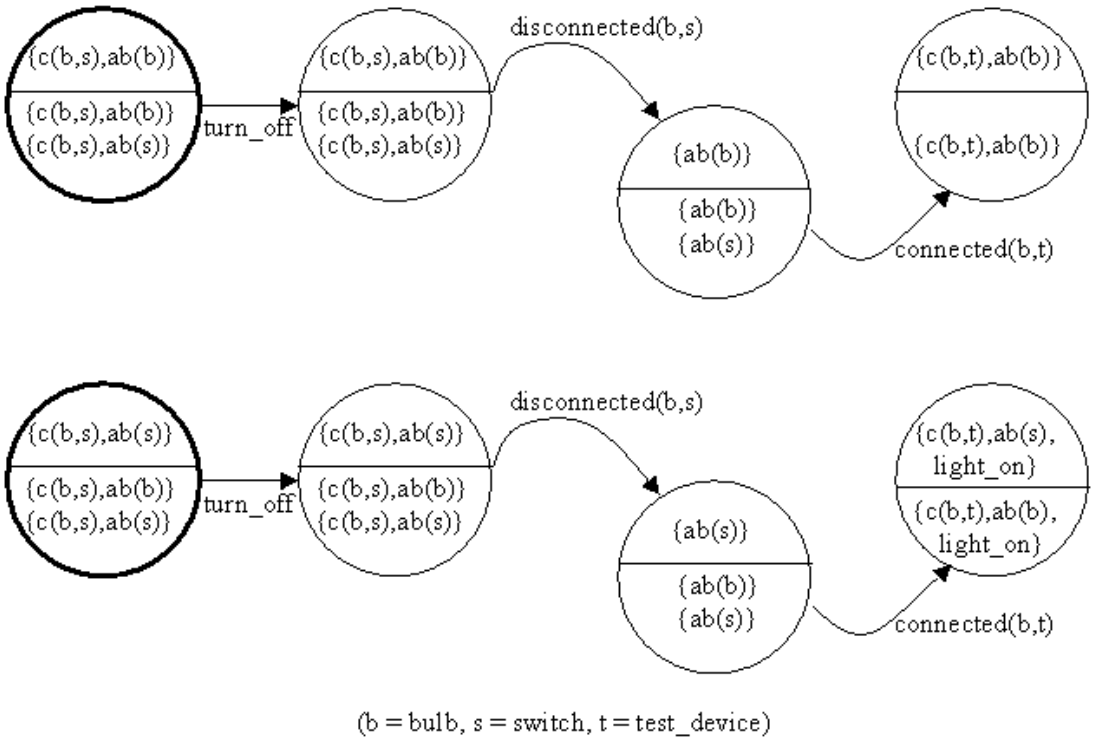
Let $\mathcal{S} = \{s_1, s_2\}$ and $L_K = \{ab(switch), ab(bulb)\}$.

Since $\hat{\Phi}([], \langle s_1, \mathcal{S}\rangle) = \{\langle s_1, \mathcal{S}\rangle, \langle s_2, \mathcal{S}\rangle\}$ and $s_1 \not\sim_{L_K} s_2$, we conclude that the empty plan is not a purely diagnostic plan for Sys_3.

We will now show that the plan

P = *turn_off* $\circ$ *disconnect(bulb,switch)* $\circ$ *connect(bulb,test_device)*

is a purely diagnostic plan for Sys_3.



The trajectories of P with respect to the two possible states are given in the above picture. We have that $\hat{\Phi}(P, \langle s_1, \mathcal{S}\rangle) = \langle s_1', \{s_1'\}\rangle$ and $\hat{\Phi}(P, \langle s_2, \mathcal{S}\rangle) = \langle s_2', \{s_2'\}\rangle$ where $s_1' = \{connected(bulb, test\_device), ab(bulb)\}$ and $s_2' = \{connected(bulb, test\_device), ab(switch), light\_on\}$.

In both cases, the values of $ab(bulb)$ and $ab(switch)$ do not change while P is executed. Furthermore, we have that $s_1 \sim_{L_K} s_1'$ and $s_2 \sim_{L_K} s_2'$. This implies that P is a purely diagnostic plan of Sys_3. □

In the next example, we consider a more complicated diagnostic plan that involves sensing actions and conditionals.

Example 9 (Electro-magnetic Door) Consider an electro-magnetic control door. The door is connected to a RED LED and a YELLOW LED. To enter, an agent needs to put its electro-magnetic card, containing its id-number and password, into the slot connected to the door's controller. The door will open only if the card is valid, the id-number and the password are not corrupted, and the door is not malfunctioning. While the card is in the slot, if it is invalid, the RED LED will be on; and if the id-number or the password is corrupted or the door is defective, the YELLOW LED will be on. The YELLOW LED is on only if the RED LED is not. In this case, pushing the button "message" will print out a message. Reading it, the agent will know whether the door is defective or its card is unreadable.

Our agent, Jack, comes to work, and as usual, puts his card into the slot. The door does not open. What is wrong ?

The story can be represented by the system $Sys_4 = (SD_4, \{card, door, id\_pwd\}, OBS_4)$ as follows.

The actions of the domain description SD_4 are: *insert_card*, *push_button*, *take_out_card*, look (look at the LEDs), or read_msg (read the message).

The fluents of SD_4 are: *ab(card)*, *ab(door)*, *ab(id_pwd)*, *has_card*, *card_in_slot*, *door_open*, *has_msg*, *red*, and *yellow*, where *red* or *yellow* indicate that the RED/YELLOW LED is on, respectively.

SD_4 comprises the following laws:

• dynamic causal laws: describing the effects of the actions *insert_card*, *push_button*, and *take_out_card*. Inserting the card causes the door to open if the card, the door and the card information are all normal. Further, inserting the card causes the card to be in the slot and not in the possession of the agent. I.e.,

$$insert\_card \textbf{ causes } door\_open \textbf{ if } \neg ab(card), \neg ab(door), \neg ab(id\_pwd)$$
$$insert\_card \textbf{ causes } \neg has\_card \wedge card\_in\_slot$$

Pushing the button results in a message. I.e.,

$$push\_button \textbf{ causes } has\_msg.$$

If the card is in the slot and the agent takes it out, then the agent has possession of the card and the card is not in the slot. I.e.,

$$take\_out\_card \textbf{ causes } has\_card \wedge \neg card\_in\_slot \textbf{ if } card\_in\_slot$$

• static causal laws: expressing the relationship between the status (on/off) of the LEDs. I.e.,

$$red \textbf{ if } ab(card) \wedge card\_in\_slot$$
$$\neg red \textbf{ if } yellow \wedge card\_in\_slot$$
$$yellow \textbf{ if } (ab(id\_pwd) \vee ab(door)) \wedge \neg red \wedge card\_in\_slot$$

• sensing actions: characterizing the knowledge effects of sensing actions. For example, performing the look action causes the agent to know whether the RED and YELLOW LEDs are on or off. They are captured by the following k-propositions:

$$\textsf{look} \textbf{ determines } red$$
$$\textsf{look} \textbf{ determines } yellow$$
$$\textsf{read\_msg} \textbf{ determines } ab(id\_pwd)$$
$$\textsf{read\_msg} \textbf{ determines } ab(door)$$

• executability conditions: characterizing when an action is precluded. I.e.,

$$\textbf{impossible } insert\_card \textbf{ if } \neg has\_card$$
$$\textbf{impossible } push\_button \textbf{ if } \neg yellow$$
$$\textbf{impossible } \textsf{read\_msg} \textbf{ if } \neg has\_msg$$

• wildcard actions:

$$break(card) \textbf{ causes } ab(card)$$
$$break(door) \textbf{ causes } ab(door)$$
$$break(id\_pwd) \textbf{ causes } ab(id\_pwd)$$

and the set of observations, OBS_4:

$$\neg red \wedge \neg yellow \wedge \neg door\_open \wedge has\_card \wedge \neg has\_msg \wedge \neg card\_in\_slot \textbf{ at } \mathsf{s}_0$$
$$insert\_card \textbf{ between } \mathsf{s}_0, \mathsf{s}_1$$
$$\neg door\_open \textbf{ at } \mathsf{s}_1$$
$$\mathsf{s}_0 \textbf{ precedes } \mathsf{s}_1$$

The first observation describes the first observable situation, s_0. The second observation states that Jack puts his card into the slot, while the third states that the door is not open after Jack puts his card into the slot.

Intuitively, when Jack observes that the door does not open as the result of putting his card into the slot, he should realize that at least one of the three components: the card, the door, or the information on the card is no longer valid. Our diagnostic reasoning system does likewise. Indeed, the narrative $(SD_4 \setminus SD_{ab}, OBS_4 \cup OK_0)$ does not have a model and there are three diagnoses for Sys_4: $\Delta_1 = \{ab(id\_pwd)\}$, $\Delta_2 = \{ab(door)\}$, and $\Delta_3 = \{ab(card)\}$ which correspond to the models M_1, M_2, and M_3 of $(SD_4, OBS_4 \cup \mathcal{D}(\{card, door\}, \{id\_pwd\}))$, $(SD_4, OBS_4 \cup \mathcal{D}(\{card, id\_pwd\}, \{door\}))$, and $(SD_4, OBS_4 \cup \mathcal{D}(\{id\_pwd, door\}, \{card\}))$ defined as follows. $M_1 = (\Psi_1, \Sigma_1)$, $M_2 = (\Psi_2, \Sigma_2)$, and $M_3 = (\Psi_3, \Sigma_3)$, where $\Psi_1([]) = s_0 \cup \{ab(id\_pwd)\}$, $\Psi_2([]) = s_0 \cup \{ab(door)\}$, and $\Psi_3([]) = s_0 \cup \{ab(card)\}$ and

$\Sigma_1(\mathsf{s}_0) = [],$
$\Sigma_1(\mathsf{s}_1) = \Sigma_1(\mathsf{s}_c) = insert\_card,$

$\Sigma_2(\mathsf{s}_0) = [],$
$\Sigma_2(\mathsf{s}_1) = \Sigma_2(\mathsf{s}_c) = insert\_card,$

$\Sigma_3(\mathsf{s}_0) = [],$
$\Sigma_3(\mathsf{s}_1) = \Sigma_3(\mathsf{s}_c) = insert\_card,$

where $s_0 = \{has\_card\}$,
$\Psi_1(insert\_card) = \{card\_in\_slot, ab(id\_pwd), yellow\} = s_1,$

$\Psi_2(insert\_card) = \{card\_in\_slot, ab(door), yellow\} = s_2,$

$\Psi_3(insert\_card) = \{card\_in\_slot, ab(card), red\} = s_3.$

To narrow the list of the possible diagnoses of the system, Jack can find out the status of the LEDs. If the RED LED is on, he knows for sure that the card is no longer valid. Otherwise, the YELLOW LED must be on. In that case, he can get the message and read it to know if the door is broken or the information on the card is corrupted. This process is captured by the following plan.

$P = look \circ$
  case
    $red \rightarrow []$
    $\neg red \rightarrow$
      case
        $\neg yellow \rightarrow []$
        $yellow \rightarrow push\_button \circ read\_msg$
      endcase
  endcase

We will now show that P is a purely diagnostic plan for Sys_4.

Let $\mathcal{S} = \{s_1, s_2, s_3\}$. There are three possible current situations of Sys_4: $\sigma_1 = \langle s_1, \mathcal{S}\rangle$, $\sigma_2 = \langle s_2, \mathcal{S}\rangle$, and $\sigma_3 = \langle s_3, \mathcal{S}\rangle$. Let $s_i' = s_i \cup \{has\_msg\}$, $i = 1, 2$, then

$\hat{\Phi}(P, \sigma_1) = \hat{\Phi}(push\_button \circ read\_msg, \langle s_1, \{s_1, s_2\}\rangle)$
$= \hat{\Phi}(read\_msg, \langle s_1', \{s_1', s_2'\}\rangle) = \{\langle s_1', \{s_1'\}\rangle\};$

$\hat{\Phi}(P, \sigma_2) = \hat{\Phi}(push\_button \circ read\_msg, \langle s_2, \{s_1, s_2\}\rangle)$
$= \hat{\Phi}(read\_msg, \langle s_2', \{s_1', s_2'\}\rangle) = \{\langle s_2', \{s_2'\}\rangle\};$

$\hat{\Phi}(P, \sigma_3) = \{\langle s_3, \{s_3\}\rangle\}$.

The above computations also represent all trajectories of P wrt σ_1, σ_2, and σ_3. Obviously, $\hat{\Phi}(P, \sigma_i) \neq \emptyset$ for $i = 1, 2, 3$. Furthermore, it is easy to check that the values of ab-literals remain unchanged in all trajectories and in each c-state $\langle s', \mathcal{S}'\rangle$ belonging to $\hat{\Phi}(P, \sigma_i)$ and $s'' \in \mathcal{S}'$, $s' \sim_{L_K} s''$ where $L_K = \{ab(card), ab(id\_pwd), ab(door)\}$. For example, $\langle s'_1, \{s'_1\}\rangle$ is the only c-state in $\hat{\Phi}(P, \sigma_1)$, and trivially, $s'_1 \sim_{L_K} s'_1$. Thus, P is a purely diagnostic plan for Sys_4. □

4.3 Repair planning

The notion of diagnostic planning discussed in the previous subsection, although does not preclude repairing of the system, focuses on getting to a unique diagnosis. Whether during executing this diagnostic plan, or after its execution, we need to repair the system so that none of its components are abnormal. That is our ultimate goal. Our aim in this subsection is to formalize this notion.

Definition 13 (Repair Plan) Given $Sys = (SD, COMPS, OBS)$. A temporal plan P of (SD, OBS) wrt. (L_M, L_W, L_K) is a *repair plan* of Sys wrt. (L_M, L_W, L_K) if $\{\neg ab(c) \mid c \in COMPS\} \subseteq L_W$.

If $L_W = \{\neg ab(c) \mid c \in COMPS\}$ and $L_M = L_K = \emptyset$, P is called a *purely repair plan.* □

We now discuss how to construct repair plans. First we discuss when all the components of the system are broken, and then we consider the alternative case and construct a repair plan from a diagnostic plan. In both our constructions, we consider only systems with independent components and repairing actions.

For constructing repair plans, we require that our system description contains laws whose effects are $\neg ab(c)$ for $c \in COMPS$. To make the matter simpler we assume that for each component c of $COMPS$, the set of laws belonging to a system $Sys = (SD, COMPS, OBS)$ contains an action

$$fix(c) \textbf{ causes } \neg ab(c)$$

and an executability condition

$$\textbf{impossible } fix(c) \textbf{ if } \neg ab(c).$$

We also assume that the process of fixing a component will not destroy another component. In other words, for every pair of states s and s' of Sys and every component c in $COMPS$ such that $ab(c) \in s$ and $s' \in \Phi(fix(c), s)$, $s \sim_{AB(COMPS \setminus \{c\})} s'$ where $AB(C) = \{ab(c) \mid c \in C\}$ for a set of components $C \subseteq COMPS$. We will refer to systems with these properties as *systems with independent components.* It is not difficult to see that for systems with independent components, the following proposition holds.

Proposition 4 For a system with independent components $Sys = (SD, COMPS, OBS)$ with $COMPS = \{c_1, \ldots, c_n\}$, the plan $P = fix(c_1) \circ \ldots \circ fix(c_n)$ is a purely repair plan for Sys iff $AB(COMPS) \subseteq s$, for every current state s of Sys, i.e., every component of the system is broken in all current states of Sys.

Proof. Let $\mathcal{S}$ be the set of current states of Sys and s be an arbitrary state in $\mathcal{S}$.

($\Longleftarrow$)
Assume that $AB(COMPS) \subseteq s$ for every $s \in \mathcal{S}$. We prove that P is a purely repair plan for Sys.

Since Sys is a system with independent components, it is easy to check that $fix(c_i)$ $(i = 1, \ldots, n)$ is executable in every c-state belonging to $\hat{\Phi}(fix(c_1) \circ \ldots \circ fix(c_{i-1}), \langle s, \mathcal{S}\rangle)$. Furthermore, due to the dynamic law $fix(c)$ **causes** $\neg ab(c)$, we have that for $c \in COMPS$, $\neg ab(c)$ is known to be true in every c-state of $\hat{\Phi}(P, \langle s, \mathcal{S}\rangle)$. Thus, $(SD, OBS) \models$ **knows** $\bigwedge_{c \in COMPS)} \neg ab(c)$ **after** P **at** s_C. (1)

It follows that $\hat{\Phi}(P, \langle s, \mathcal{S}\rangle) \neq \emptyset$ for every c-state corresponding to the current situation of Sys. Hence,

$$(SD, OBS) \models true \textbf{ during } P \textbf{ at } \mathsf{s}_C$$

and

$$(SD, OBS) \models \textbf{whether } true \textbf{ after } P \textbf{ at } \mathsf{s}_C.$$

(2)

It follows from (1) and (2) that P is a purely repair plan for Sys. (a)

($\Longrightarrow$)
Now, let P be a purely repair plan of Sys. Because $\bot \notin \hat{\Phi}(P, \langle s, \mathcal{S}\rangle)$, we have that $fix(c_1)$ must be executable in $\langle s, \mathcal{S}\rangle$. This implies that $ab(c_1) \in s$ for every $s \in \mathcal{S}$. Similarly, due to the independence of components of Sys we can conclude that $ab(c_i) \in s$ for every $i = 2, \ldots, n$, i.e., we have proved that $AB(COMPS) \subseteq s$ for every state $s \in \mathcal{S}$. (b)

The proposition is proved by (a) and (b). □

A stronger result is proved in the next proposition.

Proposition 5 For a system with independent components $Sys = (SD, COMPS, OBS)$ with $COMPS = \{c_1, \ldots, c_n\}$ and P be a purely diagnostic plan for Sys wrt. (L_M, L_W, L_K). Then, the plan $Q = P \circ P'$ with

$P' =$
Case
$ab(c_1) \rightarrow fix(c_1)$
$\neg ab(c_1) \rightarrow []$
Endcase
...
Case
$ab(c_n) \rightarrow fix(c_n)$
$\neg ab(c_n) \rightarrow []$
Endcase

is a purely repair plan for Sys.

Proof. Let $\mathcal{S}$ be the set of current states of Sys. Since P is a diagnostic plan for Sys, to prove the proposition, it suffices to show the following

(a) for every c-state $\langle s', \mathcal{S}'\rangle$ belonging to $\hat{\Phi}(P, \langle s, \mathcal{S}\rangle)$, $\bot \notin \hat{\Phi}(P', \langle s', \mathcal{S}'\rangle)$;

(b) for every c-state $\langle s', \mathcal{S}'\rangle$ belonging to $\hat{\Phi}(P, \langle s, \mathcal{S}\rangle)$ and $c \in COMPS$, $\neg ab(c)$ is known to be true in $\hat{\Phi}(P', \langle s', \mathcal{S}'\rangle)$.

Since P is a purely diagnostic plan for Sys, we know that for $c \in COMPS$, $\neg ab(c)$ is known to be true or known to be false in $\langle s', \mathcal{S}' \rangle$. Together with the fact that Sys is a system with independent components, we can conclude that P' satisfies (a) and (b). This implies that Q is a purely repair plan for Sys. □

4.4 Relationship between diagnostic planning and testing

The notion of *testing* a set of hypothesis about a system has been studied previously in the literature. Testing is intuitively related to diagnostic planning in the sense that the aim of finding a unique diagnosis (from a set of diagnosis) in diagnostic planning corresponds to treating the set of diagnosis as hypothesis and testing to find which one of them is the right one. In this subsection we first review the notion of testing in [McI97b] and then compare it to our notion of diagnostic planning in this paper.

4.4.1 Testing

We now recall the notion of testing from [McI97b]. Let Σ be a first-order theory describing the system behavior and HYP be a set of hypotheses. Each *hypothesis* $H \in HYP$ is a conjunction of distinguished literals of the language underlying Σ, whose truth value is unknown, i.e., $\Sigma \not\models H$ and $\Sigma \not\models \neg H$. There is a set AC of *achievable literals* and a set O of *observable literals*. Tests[6] are defined as follows.

Definition 14 A *test* wrt. (Σ, AC, O, HYP) is a pair (A, o) where A is a set of achievable literals and o is an observable ground literal[7]. □

In a test (A, o) the outcome of the test is o or $\neg o$.

A test (A, o) is called a *test for the hypothesis space* HYP (or test for HYP, for short) iff $\Sigma \wedge A \wedge H$ is satisfiable for every $H \in HYP$.

The outcome α of a test (A, o) *confirms* (resp. *refutes*) a hypothesis $H \in HYP$ iff $\Sigma \wedge A \wedge H$ is satisfiable, and $\Sigma \wedge A \models H \supset \alpha$ (resp. $\Sigma \wedge A \models H \supset \neg\alpha$).

A test (A, o) is an *individual discriminating test* for the hypothesis H and $\neg H$ in HYP iff $\Sigma \wedge A \wedge H'$ is satisfiable for all $H' \in HYP$ and the outcome α of (A, o) refutes either H or $\neg H$, no matter what the outcome might be.

Suppose that (A, o) is an individual discriminating test for H and $\neg H$. A tester, who knows about the test (A, o), can determine the truth value of the hypothesis H as follows. Firstly, he establishes A. Afterwards, he observes o. Depending on the truth value of o he can conclude that H is true or false, i.e., he will know the truth value of H. This is demonstrated in the next example.

Example 10 Consider a system, that describes some simple facts about a tv-set, given by

$$\begin{aligned} \Sigma \;=\; & \{power\_on \wedge picture\_on \supset tv\_ok, \\ & power\_on \wedge \neg picture\_on \supset \neg tv\_ok, \\ & picture\_on \supset power\_on\} \end{aligned}$$

[6] In this paper we only consider 'truth tests' from [McI97b].

[7] We will often omit the part "wrt. (Σ, AC, O, HYP)" when referring to a test if it is clear from the context what Σ, AC, O, and HYP stand for.

with the set of achievable literals $\{power\_on\}$ and the set of hypotheses $HYP = \{tv\_ok, \neg tv\_ok\}$. We appeal to the readers for the intuition behind the first order sentences in Σ. It is easy to see that $(\{power\_on\}, picture\_on)$ is a discriminating test for HYP. Thus, if an agent wants to know the status of the tv-set, he can turn on the TV (establishing $power\_on$) and then observes whether the picture is on or not (determining $picture\_on$). Based on the test, he knows that if $picture\_on$ is true, then the TV is ok (tv_ok is true); otherwise, the TV is broken (tv_ok is false). In other words, the agent knows the status of the TV after executing the test $(\{power\_on\}, picture\_on)$. □

The following lemma characterizes individual discriminating tests.

Lemma 1 (A, o) is an individual discriminating test for the hypotheses H and $\neg H$ in HYP iff

(i) $\Sigma \wedge A \models o \supset H$ and $\Sigma \wedge A \models \neg o \supset \neg H$

or

(ii) $\Sigma \wedge A \models o \supset \neg H$ and $\Sigma \wedge A \models \neg o \supset H$.

Proof.

($\Longrightarrow$)
Assume that (i) holds. We consider two cases:

- The outcome of the test (A, o) is o. Then, since (i) holds, from $\Sigma \wedge A \models o \supset H$ and o is true, we have that H is true. This implies that $\Sigma \wedge A \models \neg H \supset \neg o$, i.e., (A, o) refutes $\neg H$.
- The outcome of the test (A, o) is $\neg o$. Similar to the first case, we can show that (A, o) refutes H.

It follows from the above two cases that if (i) holds then (A, o) is an individual discriminating test for the hypotheses H and $\neg H$ in HYP. (1)

Similarly, we can prove that if (ii) holds then (A, o) is an individual discriminating test for the hypotheses H and $\neg H$ in HYP. (2)

($\Longleftarrow$)
Now, assume that (A, o) is an individual discriminating test for the hypotheses H and $\neg H$ in HYP.

Assume that the outcome of the test (A, o) is o. Again, we have two cases:

- (A, o) refutes H. Then, we have that $\Sigma \wedge A \models H \supset \neg o$. Since o is true, we conclude that $\neg H$ is true. It is easy to check that (ii) holds.
- (A, o) refutes $\neg H$. Similar to the first case, we can show that (i) holds.

It follows from the above two cases that if (A, o) is an individual discriminating test for the hypotheses H and $\neg H$ in HYP and the outcome is o then (i) or (ii) holds (3)

Similarly, we can prove that if (A, o) is an individual discriminating test for the hypotheses H and $\neg H$ in HYP and the outcome is $\neg o$ then (i) or (ii) holds. (4)

The lemma follows from (1)-(4). □

There is one major difference between the notion of testing and our notion of diagnostic planning. In diagnostic planning there may be restrictions on changing the values of the hypotheses during its execution whereas there are no such restrictions in the notion of testing. The following example illustrates this point.

Example 11 Consider a system, whose behavior is specified by

$$\Sigma = \{l \wedge o \supset \neg h, l \wedge \neg o \supset h\}$$

with the hypothesis space $HYP = \{h, \neg h\}$, the set of achievable literals $AC = \{l\}$, and the set of observable literals $O = \{o, \neg o\}$. We will now show that $(\{l\}, o)$ is an individual discriminating test for h and $\neg h$.

First we show that $\Sigma \wedge \{l\} \wedge H$ is satisfiable for all $H \in HYP$. This follows immediately from the fact that $\{l, o, \neg h\}$ is a model for $\Sigma \wedge \{l\} \wedge \neg h$, and $\{l, \neg o, h\}$ is a model for $\Sigma \wedge \{l\} \wedge h$. So, we have that $\Sigma \wedge \{l\} \wedge H$ is satisfiable for every H in HYP. (1)

We now need to prove that $(\{l\}, o)$ refutes either h or $\neg h$, no matter what the outcome of $(\{l\}, o)$ might be. There are two cases:

1. The outcome of $(\{l\}, o)$ is o. It follows from $(o \supset \neg h) \wedge (\neg o \supset h) \models h \supset \neg o$ that $\Sigma \wedge \{l\} \models h \supset \neg o$. Thus, the test refutes h. (2)

2. The outcome of $(\{l\}, o)$ is $\neg o$. Since $(o \supset \neg h) \wedge (\neg o \supset h) \models \neg h \supset o$ we conclude that $\Sigma \wedge \{l\} \models \neg h \supset o$. So, the test refutes $\neg h$. (3)

It follows from (1), (2), and (3) that $(\{l\}, o)$ is an individual discriminating test for h and $\neg h$.

Consider the case that initially the world is described by the model $M = \{\neg l, o, h\}$. But the agent is not aware of that and intends to test whether h or $\neg h$ holds. For that, as required he would need to establish l. But then the model of the world could change to $M' = \{l, o, \neg h\}$[8]. *Thus, while establishing l which is needed for the testing, the tester may change the value of the hypothesis h.* □

Now if the hypotheses were fluents about the abnormality of components, we may not want their values to be changed during testing. That is because, in the extreme case the agent can just break the whole system and claim to have determined that all components are abnormal. To avoid this in our notion of diagnostic planning we have the provision to declare these fluents about abnormality of components to be *protected*. No such provisions are part of the original notion of testing.

In the next definition, we formalize precisely what is a test which maintains a set of literals.

Definition 15 A test (A, o) *maintains* a literal l in Σ if for every model M of Σ, the following conditions hold:

(i) for every model M' of $\Sigma \cup A$, which is closest to M, $M \sim_l M'$[9];

(ii) there exists a model M' of $\Sigma \cup A$ satisfying (i).

(A, o) maintains a set of literal L if it maintains every element l of L.

In what follows, we will show that each individual discriminating test for hypotheses H and $\neg H$, which maintains the set of hypotheses HYP, can be viewed as a discriminating diagnostic plan, which determines the truth value of H.

[8] Another possibility would be $\{l, \neg o, h\}$.

[9] $M \sim_l M'$ means that l is true in M iff l is true in M'.

4.4.2 Testing vs diagnostic planning

Let T be a testing domain, given by Σ, HYP, AC, and O. For simplicity, we assume that HYP is a set of literals such that if $H \in HYP$ then $\neg H \in HYP$. It is easy to see that T can be represented by the system description $Sys^T = (SD, COMPS, OBS)$ where

- SD consists of
  - the set of static causal laws $\tilde{SD} = \{\varphi \textbf{ if } true \mid \varphi \in \Sigma\}$ and
  - the set of dynamic causal laws $\{make(A) \textbf{ causes } l \mid l \in A, A \subseteq AC\}$;
- $OBS = \emptyset$; and
- $COMPS = \{H \mid H \text{ is a positive hypothesis in } HYP\}$.

Furthermore, the set of observable literals of SD is O. Since our action language does not have a notion of achievable literals, to capture the fact that AC is the set of achievable literals of T, we require that for every initial state s of Sys^T, the following condition is satisfied

(*) for each set of achievable literals A of T, there exists a state s' of Sys^T such that $Cn(s') = Cn(A \cup (s \cap s') \cup \tilde{SD})$.

In the following, we will consider only testing domains, whose corresponding system descriptions satisfy (*). We believe that this is a reasonable restriction since it is equivalent to the assumption that achievable literals can be established at any time. The next proposition shows that individual discriminating tests are special case of diagnostic plans.

Proposition 6 Let T be a testing domain and Sys^T be its corresponding system description. Then, if (A, o) is an individual discriminating test for H and $\neg H$ maintaining HYP, then $make(A)$ is a discriminating diagnostic plan of Sys^T for H.

Proof. Let $\mathcal{S}$ be the set of possible states of SD and s be an arbitrary state in $\mathcal{S}$. Since Sys^T satisfies (*), for every $s' \in \mathcal{S}$, $\Phi(make(A), s') \neq \emptyset$. Hence, $\hat{\Phi}(make(A), \langle s, \mathcal{S}\rangle) \neq \emptyset$. (1)

As (A, o) maintains HYP, it is easy to see that values of literals in HYP remain unchanged in all trajectories of $make(A)$ wrt. any possible states of Sys^T, i.e., $(SD, OBS) \models HYP$ **during** P **at** s_C. (2)

Consider a c-state $\langle s', \mathcal{S}'\rangle$ in $\hat{\Phi}(make(A), \langle s, \mathcal{S}\rangle)$. Assume that s'' is a state in $\mathcal{S}'$. It follows from the definition of Φ that $Cn(s') = Cn(A \cup (s' \cap s) \cup \tilde{SD})$ and $Cn(s'') = Cn(A \cup (s'' \cap s^+) \cup \tilde{SD})$ for some $s^+ \in \mathcal{S}$. Furthermore, $s' \sim_O s''$. In other words, s' and s'' are models of $\Sigma \wedge A$ which agree on O. It follows from the assumption that (A, o) is an individual discriminating test for H and $\neg H$ that $s' \models H$ iff $s'' \models H$. This implies that $s'' \sim_H s'$. Since (A, o) maintains HYP and s and s' are models of Σ and $\Sigma \wedge A$ respectively, we have that $s \sim_{HYP} s'$. Together with $s'' \sim_H s'$, we conclude that $s \sim_H s''$. (3)

It follows from (1)-(3) that $make(A)$ is a discriminating diagnostic plan of Sys^T for H. □

The above proposition shows that the notion of discriminating diagnostic planning does extend the notion of individual discriminating testing. In the next example, we show that the consistency requirement in the formalization of a test might be too strong and unnecessary.

Example 12 Let us consider the testing domain $T = (\Sigma, HYP, AC, O)$ with

$\Sigma = \{l \supset \neg h_2, l \wedge \neg h_1 \supset o, l \wedge h_1 \supset \neg o\}$ and $AC = \{l\}$, $HYP = \{h_1, h_2, \neg h_1, \neg h_2\}$, and $O = \{o\}$.

It is easy to see that the plan to achieve l (i.e., the plan $make(\{l\})$) is a discriminating diagnostic plan of Sys^T for h_1. However, $(\{l\}, o)$ is not an individual discriminating test for h_1 because $\Sigma \wedge \{l\} \wedge h_2$ is unsatisfiable. □

In the next proposition, we show that under certain conditions a discriminating diagnostic plan can be translated into an individual discriminating test. More precisely, if O is singleton (the set of observations of T consists of only one observable literal), $make(A)$ is a discriminating diagnostic plan for H, and the consistency condition $\Sigma \wedge A \wedge H'$ is not violated for any hypothesis H' then we can construct an individual discriminating test for H and $\neg H$.

Proposition 7 Let $T = (\Sigma, HYP, AC, \{o\})$ be a testing domain, Sys^T be its corresponding system description, and $make(A)$ be a discriminating diagnostic plan of Sys^T for H. Assume that $\Sigma \wedge A \wedge H'$ is satisfiable for every $H' \in HYP$. Furthermore, assume that $\Sigma \wedge A \wedge o$ and $\Sigma \wedge A \wedge \neg o$ are satisfiable. Then, (A, o) is an individual discriminating test for the hypotheses H and $\neg H$.

Proof. First, by the assumption of the lemma, for every $H \in HYP$, $\Sigma \wedge A \wedge H$ is satisfiable. (1)

Consider a model s_0 of Σ such that $s_0 \models \Sigma \wedge A \wedge o$. There are two cases:

(a) $s_0 \models \neg H$. We will prove that

(*) for each model s of $\Sigma \wedge A \wedge o$, $s \models \neg H$.

Assume the contrary, i.e., (*) does not hold. This mean there exists a model s_1 of $\Sigma \wedge A \wedge o$ such that $s_1 \models H$.

Let $\mathcal{S}$ be the set of possible states of Sys^T. Consider the c-state $\langle s_0, \mathcal{S} \rangle$. Because s_1 is a model of Σ and each model of Σ is a possible state of Sys^T and vice versa, we have that $s_1 \in \mathcal{S}$. Since $s_0 \cap s_1 \models \Sigma \wedge A \wedge o$, we can conclude that $s_1 \in \Phi(make(A), s_0)$. Thus, there exists a c-state $\langle s_1, \mathcal{S}' \rangle$ in $\Phi(make(A), \langle s_0, \mathcal{S} \rangle)$ and $s_1 \not\sim_{HYP} s_0$. This implies that $make(A)$ is not a discriminating diagnostic plan of Sys^T for H. Hence, our assumption is incorrect, i.e., we have proved (*). Since $\Sigma \wedge A \models H \supset \neg o$ iff $\Sigma \wedge A \models \neg H \vee \neg o$, it follows from (*) that no matter what the outcome of (A, o) is, $\Sigma \wedge A \models H \supset \neg o$ holds. Thus, (A, o) refutes H. (2)

(b) $s_0 \models H$. In this case, similar to the proof of the first case, we can prove the following:

(**) for each model s of $\Sigma \wedge A \wedge \neg o$, $s \models H$.

It is easy to see that (**) implies that $\Sigma \wedge A \models \neg H \supset \neg o$, which means that (A, o) refutes $\neg H$. (3)

The conclusion of the proposition follows from (1)-(3). □

5 Related Work

In this section, we contrast our contributions to related work. In the area of diagnosis of dynamical systems, there has been research both within the control theory community (e.g., [SSL+96]) on the diagnosis of discrete event systems using finite state automata, and within the AI community. Most of this work is fairly recent, and can be differentiated with respect to the expressive power of the language used to model the domain (e.g., propositional/first order, ramifications, nondeterministic actions, concurrent actions, narrative, sensing, probabilities); how the notion of diagnosis is defined (e.g., models, sequences of actions, sets of abnormal components, probabilistic criteria); how observations are expressed; whether diagnosis is active or offline; and what aspects of diagnostic problem solving, beyond diagnosis, are addressed (e.g., diagnostic planning, repair). We now discuss more in details the relationship between our work and several related work on dynamic diagnosis within the AI community. We start with a comparison to McIlraith's work as our work in this paper is very closely related to the research in McIlraith's thesis [McI97b].

5.1 Relation with McIlraith's work

Our work was influenced by previous work of McIlraith (e.g., [McI97a, McI98, McI97b]), but extends and builds on aspects of that work in several important ways. [McI97b] argued that a comprehensive account of diagnostic problem solving must involve reasoning about action and change, and provided such an account in a dialect of the situation calculus that included causal ramification constraints, but did not include nondeterministic actions, sensing actions or narrative. Aberrant behavior was assumed to be caused by unobserved exogenous actions. Multiple definitions of diagnosis were provided both in terms of sequences of actions that explained the observations, and designations of normal and abnormal components with respect to a situation. The notion of a diagnostic model was not employed. Most importantly, this account did not exploit narrative for expressing and accounting for observations, consequently the assertion of observations and exogenous actions was much less elegant. [McI97b] also introduced the notion of testing to discriminate hypotheses, and analogues to the ideas of diagnostic and repair planning; however since the dialect of the situation calculus she employed did not include knowledge-producing actions, the important integration of sensing and world-altering actions that was done in this paper, was argued for but was left to future work.

For a precise comparison between the two formalisms, we now recall her definitions for consistency-based diagnosis, abductive diagnosis and explanatory diagnosis.

Definition 16 (Consistency-based diagnosis) Given a system $(\Sigma, HIST, COMPS, OBS_F)$, where Σ is a set of situation calculus sentences describing the behavior of the system and the actions that can affect it, $HIST$ is a sequence of ground actions that were performed starting in s_0, $COMPS$ is a finite set of constants, and OBS_F is a fluent formula.

Suppose there exists a sequence of ground actions , s_HIST such that $\Sigma \wedge Poss(HIST.s\_HIST, \mathsf{s}_0)$ is satisfiable.

Further, let $\mathsf{s} = do(HIST.s\_HIST, \mathsf{s}_0)$.

An AB-hypothesis $\mathcal{D}_\mathsf{s}(\Delta_1, \Delta_2) = \bigwedge_{c \in \Delta_1} AB(c, \mathsf{s}) \wedge \bigwedge_{c \in \Delta_2} \neg AB(c, \mathsf{s})$ is a consistency based diagnosis for our system, relativized to situation s iff:

- $\Delta_1 \cup \Delta_2 = COMPS$, and

- $\Sigma \wedge OBS_F(\mathsf{s}) \wedge \mathcal{D}_\mathsf{s}(\Delta_1, \Delta_2) \wedge Poss(HIST.s\_HIST, \mathsf{s}_0)$ is satisfiable. □

She then defines an ordering between action sequences, where an action sequence α is defined to be simpler than an action sequence β if (i) actions in $\alpha \subseteq$ actions in β, and (ii) length of α is smaller than length of β.

Definition 17 $\mathcal{D}_\mathsf{s}(\Delta_1, \Delta_2)$ is a chronologically simple consistency-based diagnosis for a system, if it is a consistency-based diagnosis for the system and there is no consistency-based diagnosis $\mathcal{D}_{\mathsf{s}'}(\Delta_1', \Delta_2')$ such that the action sequence in s' is simpler than the action sequence in s. □

Her definition of abductive diagnosis differs from the consistency-based diagnosis in the following respect:

An AB-hypothesis is a consistency-based diagnosis if it is consistent with the theory and the observation, while it is an abductive diagnosis if when conjoined to the theory it actually entails the observations, while the theory alone does not.

In addition if the requirement that $\Delta_1 \cup \Delta_2 = COMPS$ is removed then she refers to the AB-hypothesis that entail the observations as an abductive explanation.

Definition 18 (Assumption based explanatory diagnosis) Given an assumption $H(\mathsf{s})$ relativized to ground situation s such that

- $\mathsf{s}_0 \leq \mathsf{s} \leq do(HIST.E, \mathsf{s}_0)$,
- $\Sigma \wedge H(\mathsf{s})$ is satisfiable, and
- $\Sigma \wedge H(\mathsf{s}) \models Poss(HIST, \mathsf{s}_0)$

an assumption based explanatory diagnosis for a system $(\Sigma, HIST, COMPS, OBS_F)$ under assumption $H(\mathsf{s})$ is a sequence of actions $E = [\alpha_1, \ldots, \alpha_k]$ such that

$\Sigma \wedge H(\mathsf{s}) \models Poss(HIST.E, \mathsf{s}_0) \wedge OBS_F(do(HIST.E, \mathsf{s}_0))$ □

We now list the differences between our formalism and McIlraith's notions of diagnosis.

1. In all her definitions the observations are about a single situation. We allow observations about different situations.

2. Her formulation assumes a given history of action execution starting from the initial situation s_0, and observations about a single later situation, and her notion of diagnosis centers around looking for action sequences that might have happened between the known history and the observation point. In our approach we allow the possibility that actions may have happened during what we know of the history. Thus, while she forces the missing actions to occur at the end of the observed history, in our approach the missing actions could be intermixed anywhere in the narrative.

3. Her definitions of diagnosis, although uses the situation calculus action language, is outside the language of situation calculus. For example, in her definition of consistency based diagnosis OBS_F is not directly added to Σ, rather it is relativized with respect to a situation s, *that is guessed.* (In other words, her observations OBS_F is not in the language of situation calculus.)

In our formulation the observations are in the action language we use and are directly added to the action theory.

The difference between the two approach is analogous to the difference between planning within the language (as in Situation calculus) and planning outside the language using abduction in event calculus or a similar language. Reiter in [Rei96] compares the two approaches and describes the advantage of the first approach over the second.

4. The difference described in the previous items partly stems from the difference in the action language that we use and the one she uses. Our action language is meant for narratives and thus observations at different time instances are describable within the language, while the version of Situation calculus she uses does not allow narratives. We would like to add that Pinto and Reiter do have extensions [PR93, Pin94] that allow narratives, but there language, as described in [PR93, Pin94] suffers from the problem of premature minimization[10] and is not able to do planning, which becomes important when we get to diagnostic planning and repair. (In [BGP98] we have a detailed comparison of our action language for narratives and other such languages in the literature.)

5. Our notion of diagnosis integrates the various aspects such as (i) assumption that initially all components are normal, (ii) minimization of missing action occurrences with respect to the subsequence relationship into one definition. The first assumption is missing in her definition of consistency-based diagnosis and abductive diagnosis, and can be formulated in her assumption based explanatory diagnosis. The minimization of action sequences done in her chronologically simple diagnosis is done on top of the definitions of diagnosis, while in our case the minimization is part of the semantics of the action language.

6. In her definitions of diagnosis, in case of consistency based diagnosis and abductive diagnosis, the diagnosis is an AB-hypothesis while in case of explanatory diagnosis it is a sequence of actions. In our definition of diagnosis (Definition 3) we define the notion of a *diagnostic model* from which we can not only find the missing action occurrences (similar to her explanatory diagnosis), but also find the AB-hypothesis at different situations. (The AB-hypothesis with respect to the current situation would correspond to her consistency-based diagnosis.)

   Our formulation of a diagnostic model is basically consistency-based. But, if we assume that actions are deterministic and we have complete information about the initial situation then it is also an abductive diagnosis. This is not the case with her formulation of consistency-based diagnosis. That is because her notion of a consistency based diagnosis is an AB-hypothesis, not a set of action occurrences; and there may exist action sequences that which result in the same AB-hypothesis but are inconsistent with the observations.

In Section 4.4 we show some relationship between the notion of testing in [MR92] and our notion of diagnostic planning. We now discuss additional differences between the two:

1. She defines a test as a pair (A, o) where A is a set of achievable literals, and o is a ground observable, and the main intuition is that the agent should achieve A and then observe o, to narrow down the set of possible diagnosis (or any other set of hypotheses in general).

[10] Reiter ..

Her definition of achieving a test can be intuitively described as a sequence of actions which when executed in any of the current situations (as dictated by the various possible diagnosis) results in a situation where A holds.

Our notion of a diagnostic plan incorporates the above two aspects into a single framework. *In addition we use sensing actions.* (In fact in Page 141 of her thesis [McI97b] she lists such an integration using sensing actions as one of her future work.)

2. The other main difference, which we pointed out earlier in Section 4.4 is that we do not allow actions during the testing or diagnostic planning to permanently alter the value of certain fluents, in particular the *ab* fluents, determination of whose value is one of the reason behind doing the diagnosis in the first place.

3. Regardless of the above two differences, we believe that her notion of testing is an extremely important contribution, and plays the same role with respect to our notion of diagnostic planning as the original notion of diagnosis in [Rei87] plays with respect to notions of diagnosis with actions and narratives in [McI97b] and Section 3 respectively.

In Chapter 6 of her thesis she also defines a repair plan as a sequence of actions which when executed in any of the current situations (as dictated by the various possible diagnosis) results in a situation where a certain repair objective R holds. We allow sensing actions, and hence conditional statements in our repair plans, and articulate a repair objective to be reaching a state where all components of the system are fine.

5.2 Relation with Thielscher

Thielscher in [Thi97] characterizes diagnoses in terms of minimally failing components, where his minimization preference criterion is with respect to the abnormalities in the initial state, but can additionally exploit some a prior likelihood. He does not exploit exogenous actions to account for abnormalities as we do, and does not allow for the occurrences of actions beyond what are observed. Also, he does not take his work beyond a characterization of diagnosis.

Definition 19 A system description Sys is a tuple $(\epsilon, \mathcal{F}, \mathcal{A}, \mathcal{L}, C, R, <)$, consisting of entities, fluents, action names, action laws, state constraints, causal relationships, and a partial ordering on the set of ground instances of $ab \in \mathcal{F}$. □

In his formulation, a diagnosis problem is a pair (Sys, O), where Sys is a system description and O is a set of observations of the forms f **after** $[a_1, \ldots, a_n]$. An interpretation for a diagnosis problem is a pair (Res, Σ), where Res maps action sequences to states, and Σ maps state, action pairs to a set of states. An interpretation is said to be a model if it satisfies (in an intuitive way) the action laws, state constraints, causal relationships and the observations.

Definition 20 Let (Sys, O) be a diagnosis problem with partial ordering $<$. If $M = (Res, \Sigma)$ is a model for (Sys, O) then M is preferred iff we can find a strict ordering $\ll$ extending $<$ such that the following holds:

For each model $M' = (Res', \Sigma)$ for (Sys, O) and each fluent $ab(c) \in Res([]) \setminus Res'([])$, $\exists ab(c') \in Res'([]) \setminus Res([])$ such that $ab(c) \ll ab(c')$. □

In his formulation diagnoses are extracted from the preferred models. The main similarities and differences between his and our approaches are as follows:

1. His Σ and *Res* is exactly same as our *Res* and Ψ respectively. But since we allow narratives, we have in addition a mapping from situations to action sequences.

2. His formulation does not allow occurrences of actions beyond what is observed. In other words his observations are not explained in terms of what actions (possibly exogenous) may have occurred unnoticed by the observer. This is the main concept in our approach. In the ETAI discussions of [Thi97] Cordier and an anonymous reviewer also point this out with respect to Thielscher's formulation. In response Thielscher says that "The happening of a (non-exogenous) event is considered a property of a world state and thus is formally treated as a fluent. Hence it can be (directly or indirectly) caused by actions or other events. ...". From this response it is not clear to us how observations can be explained using missing actions.

3. He uses minimization (or a preference criteria) with respect to *ab* fluents in the initial state. Our preference is with respect to minimizing missing action occurrences with respect to subsequence relationship.

4. We use sensing actions and conditional plans to go beyond diagnosis and formulate diagnostic planning and repair.

5.3 Relation with Cordier

Another important piece of work from the AI community is the work of Cordier, Thiébaux and their co-authors, (e.g., [TCJK96, CT94]). Their work is similar in spirit to ours, viewing the diagnosis task as the determination of an event-history of a system between successive observations. While this work is related, the representation of the domain uses state transition diagrams and is much less expressive and elaboration tolerant than ours. That said, their representation system is sufficiently expressive for the power distribution domain they have been examining, and more recently, their work has focused on the necessary tradeoffs required to address hard computational issues associated with their domain. Cordier and Thiébaux also discuss the notion of repair planning, but without distinguishing between sensing and world-altering actions. We now give formal definitions of her notions.

Definition 21 A systems is a pair $(Sys, \mathcal{E})$, where Sys is a set of states (representing the possible states of the system) and $\mathcal{E}$ is a set of events, where an event is a relation on $Sys \times Sys$. □

She represents an observation OBS as a subset of Sys. By a scenario she refers to a sequence of events, and a scenario $[e_1, \ldots, e_n]$ for a system $(Sys, \mathcal{E})$ denotes the following relation on $Sys \times Sys$:

$$\begin{cases} \{(s_1, s_{n+1}) \mid \exists s_2 \in Sys, \ldots, \exists s_n \in Sys \ s.t. \ \forall i \{1, \ldots, n\}(s_i, s_{i+1}) \in e_i\} & if \ n \geq 1 \\ \{(s, s) \mid s \in Sys\} & otherwise \end{cases} \tag{15}$$

Definition 22 An event based diagnosis for $(SYS, [OBS_1, \ldots, OBS_m])$, $m \geq 2$, is a scenario E for SYS such that: $E = E_1 \bullet E_2 \ldots \bullet E_{m-1}$, and $\exists s_1 \in OBS_1, \ldots, \exists s_m \in OBS_m$ such that $\forall i \in \{1, m-1\}(s_i, s_{i+1}) \in E_i$. □

She then defines several preference relations between scenarios and uses them to define corresponding preferred event based diagnosis: most probable diagnoses, shortest diagnoses, minimal diagnoses and least redundant diagnoses.

Definition 23 An explanatory event based diagnosis for $(SYS, [OBS_1, OBS_2])$ is an event based diagnosis E for $(SYS, [OBS_1, OBS_2])$ such that $\forall s_1 \in OBS_1 \not\exists s_2 \in Sys \setminus OBS_2$ such that $(s_1, s_2) \in E$.

An explanatory event based diagnosis for $(SYS, [OBS_1, \ldots, OBS_m])$ $m \geq 2$, is an event based diagnosis E for $(SYS, [OBS_1, \ldots, OBS_m])$ such that $E = E_1 \bullet \ldots \bullet E_{m-1}$, where $\forall i \in \{1, \ldots, m-1\}$ $E_1 \bullet \ldots \bullet E_i$ is an explanatory event-based diagnosis for $(SYS, [OBS_1, OBS_i])$ □

Our approach is similar to Cordier's in the sense that in both approaches observations are explained by missing action occurrences. The differences between our formalism and hers are as follows:

1. Our observations include action occurrences at different time points. Her observations are only about fluent values at different time instances.

2. In our formulation we assume that all ab fluents are false (i.e., all components are normal) in the initial situation. She does not make such assumptions, although such an assumption can be formalized by including them as part of OBS_1. If we do not worry about the ab fluents then explaining observations (as done in Cordier's) is a feature of our language $\mathcal{L}$ in our earlier work in [BGP98].

3. The main difference between the two approaches is our use of an domain description language, while her formulations are based on the state transition diagram. The main advantage of using an appropriate domain description language is the ease and elaboration tolerance [MH69] of the specification of actions and their effects, and the constraints about the world. In the ETAI discussions about his paper [Thi97] Thielscher elaborates on this issue when responding to Cordier's comments.

4. Finally, diagnosis is only one of our concerns in this paper. Our main goal is to give a uniform formulation of diagnosis, diagnostic planning and repair. She does not discuss diagnostic planning and repair, and in particular she does not consider sensing actions and conditional (possibly knowledge gathering) plans.

5.4 Relation with temporal diagnosis formulations

In [BCTD98] Brusoni et al. present a spectrum of definitions for temporal model-based diagnosis based on three main dimensions:

1. temporal phenomena (with three aspects);

   (a) time varying context (different observations at different time points);

   (b) temporal behavior (discrete state change vs using qualitative derivatives); and

   (c) time varying behavior (different faults at different times).

2. ontology of time (metric time vs qualitative time vs time as a sequence of states vs use of ad-hoc abstract temporal primitives); and

3. definition of diagnosis (full spectrum between consistency based diagnosis and abductive diagnosis).

They discuss the appropriateness of the different definitions in the spectrum for different cases and analyze how different approaches to temporal diagnosis in the literature (such as [Str97]) can be cast into their framework.

They introduce a modeling language which can be used to express the various temporal phenomena, and which is independent of the ontology of time. In their formulation, the behavior of the system is described by two sets of formulas: a set of temporal behavior formulae; and a set of temporal integrity constraints.

Their temporal behavior formula is of the form:

$$a_1(X_1, T_1), \ldots, a_n(X_n, T_n) \textbf{ explains } b_1(Y_1, T'_1), \ldots, b_m(Y_m, T'_m)\{C(T_1, \ldots, T_n, T'_1, \ldots, T'_m)\}$$

where $a_i(X_i)$ and $b_j(Y_j)$ denote conditions on the system being modeled whose truth value may change over time, T_i and T_j's denote time intervals where $a_i(X_i)$ and $b_j(X_j)$ are true, and $C(T_1, \ldots, T_n, T'_1, \ldots, T'_m)$ is a set of temporal constraints.

Following is an example of a temporal behavior formula.

$$oil\_loss(T_{ol}) \textbf{ explains } low\_oil\_pressure(T_{lop})\{T_{ol} \ overlaps \ or \ meets T_{lop}\}$$

Their temporal integrity constraint is a logical formula of temporal constraints involving maximal episodes for some entities in the model. An example of a temporal integrity constraint is:

$$a(v_1, T_1) \wedge a(v_2, T_2) \rightarrow T_1 \ disjoint \ T_2$$

Their modeling language is analogous to our high level language $\mathcal{L}$, and their 'explains' has a similar meaning as our 'causes'. In general the main differences between their approach and ours are:

1. They do not have an explicit notion of an action. We distinguish between two kinds of 'causes': dynamic causal laws (an action causing a fluent) and static causal laws (a fluent causing a fluent). At first glance it seems that they do not have dynamic causal laws and their temporal behavior formulas are like our static causal laws. But in the example that we borrow from their paper, *oil_loss* is an action. So it seems that they treat actions also as fluents.

2. They do not discuss diagnostic planning and repair, and do not consider sensing.

3. As presented in this paper, we have a fixed notion of time. We model time using situations, and our situations are not sequences of states, rather they record the sequence of actions since the initial situation. (Recall that situations are characterized by mappings from situations to action sequences.) Thus our notion of time is not among the time ontology discussed in their paper.

   On the other hand they discuss several different way to model time. Even though we have a fixed notion of time, we could extend our approach to also assign a starting and end time to each situation. (Such a formulation is done in [Rei96] and we can adopt it to our formulation.)

In general the main difference in their work and ours is our focus on a unifying framework for diagnosis, diagnostic planning, testing and repair, with 'actions' as the central theme. To avoid distraction from our main point we settle on a much narrower spectrum than they. In particular, our formulation can express time varying context, time varying behavior, discrete state change

aspect of temporal behavior, situation notion of time, and various definitions of diagnosis; and can not express qualitative derivative aspect of temporal behavior, metric time, qualitative time, and some ad-hoc primitives.

The above differences also mostly hold with respect to the many temporal model based diagnosis papers discussed in [BCTD98]. In addition, while our formulation is based on discrete transitions due to actions, some of the other temporal MBDs such as [Str97] consider continuous change. We would like to add that incorporating continuous change to our framework can be accomplished – without abandoning the notion of actions – by following the techniques in [Rei96, Pin94], where they have discrete actions that start and halt a process and in between the process goes through continuous change.

Williams and Nayak in [WN96, WN97] and Muscettola et al. [MNPW98] discuss model based diagnosis and repair in the space craft domain. They use concurrent transition systems to model systems and use a subset of the temporal logic defined in [MP92] to specify them. The main parts of their system related to our work are: MI (mode identification), MR (mode reconfiguration), and MRP (model based reactive planner). The MI is the diagnostic component which infers the current state from (partial) knowledge of the previous state (and the history), and current (and past) observations. Thus their notion of diagnosis differs from ours, as in our case we are also interested in determining the missing action occurrences that explain our observations. In [WN96], the MR uses the output of MI to determine a set of control values (i.e., a set of actions to be executed concurrently) such that regardless of which is the real current state, the next state the system reaches satisfies the configuration goal. In [WN97], the MR uses the output of MI to just determine a reachable target state that satisfies the goal configuration, and the MRP generates the first action in a control sequence for moving from the most likely current state to the target state. Neither of their MR or MRP involves planning with sensing actions as their observations are not directed, rather they make all the observations that are possible in each state.

5.5 Relation with work on diagnostic and repair planning

In the area of diagnostic and repair planning, Sun and Weld [SW93] proposed a decision-theoretic planner which was invoked by a diagnostic reasoner to plan repair actions. The associated planning language distinguished between information-gathering and state-altering actions, but did not provide for the specification of knowledge or diagnostic goals. In their approach the diagnosis code maintains a model of the most probable modes of the device's components and uses the planner and certain cost and utility analysis to suggest useful action sequences. The main differences between their work and ours is that their notion of diagnosis is not with respect to a narrative and is not in terms of missing action occurrences. It seems that they integrate planning (that generate action sequences) and execution while we consider conditional plans with sensing actions. In particular, we take advantage of recent advances in reasoning about actions (done after their paper) in terms of using action theories that can reason and explain narratives [BGP98], action theories that allow causal statements [Tur97], and action theories that allow sensing actions and can reason with conditional plans [Lev96, BS98, DL99].

Similarly Heckerman et al. [HBR94] have examined the problem of interactively generating repair plans under uncertainty using Bayes nets, a single fault assumption and a myopic lookahead heuristic. Actions are limited to simple observations and component replacement. In contrast Friedrich et al. (e.g., [FN92]) developed a set of greedy algorithms to choose between performing simple observations and repair actions, assuming a most likely diagnosis. They do not limit their system

to repair alone but rather generalize their goal to some notion of purpose; purpose does not include specification of diagnostic goals. Finally, and perhaps most notably, Rymon [Rym93] developed a goal-directed diagnostic reasoner and companion planner, called TraumAID 2.0. The primary task of the diagnostic reasoner was to generate goals for the planner and to reason about whether those goals were satisfied.

6 Conclusions

In this paper we provided an account of diagnostic problem solving in terms of the action language, $\mathcal{L}$. A prime objective of this work was to characterize diagnostic problem solving with narrative and sensing. $\mathcal{L}$ proved ideal for this task because it already had most of the necessary expressive power. In particular, $\mathcal{L}$ includes narrative. In this paper, we extended $\mathcal{L}$ by adding static causal laws and sensing actions that are necessary for describing the behavior of the systems we diagnose. To the best of our knowledge ours is the first action theory that allows both sensing and narrative. We also distinguished notions of observable fluents, and protected, restored and fixable fluents.

The main contributions of this paper, in addition to the supporting language extensions, are the characterization of the diagnosis task as a narrative understanding task, and the definition of diagnosis in terms of a diagnostic model – a particular model of the narrative. We further distinguish between a diagnostic model and the stricter notion of an explanatory diagnostic model. As discussed throughout the paper, diagnostic problem solving is more than just determining a set of candidate diagnoses. In the second half of the paper, we define the notion of a diagnostic plan, and a repair plan – conditional plans that exploit both world-altering actions and sensing actions with the goal of achieving some diagnostic knowledge or repair objective. These present new contributions to the research on model-based diagnosis and reasoning about action. In the future, we plan to develop an engine for diagnostic problem solving and discuss complexity issues related to computing current diagnoses, constructing diagnostic and repair plans.

Appendix

The next lemmas are used to prove the equivalence between minimal current fluent diagnoses for SD' and diagnoses for SD.

Lemma 2 If Δ is a diagnosis for Sys then Δ is a current fluent diagnosis for Sys'.

Proof. Let Δ be a diagnosis for Sys. We construct an $M = (\Psi, \Sigma)$ as follows. $\Sigma(\mathsf{s}_C) = \Sigma(\mathsf{s}_1) = make(OBS)$. By Lemma 4, there exist two states of Sys', s_1 and s_2, which satisfy (a), (b), and (c) of the Lemmma 4. Define $\Psi([]) = s_1$ and $\Psi(make(OBS)) = s_2$. It follows from (a)-(c) of Lemma 4, $s_2 \in \Phi(make(OBS), s_1)$. Hence, M is a model of $(SD', OBS' \cup OK_0)$. Since s_2 satisfies $SD \cup OBS \cup \{ab(c) \mid c \in \Delta\} \cup \{\neg ab(c) \mid c \in COMPS \setminus \Delta\}$, we conclude that $ab(c)$ **at** s_C (resp. $\neg ab(c)$ **at** s_C) holds in M for $c \in \Delta$ (resp. $c \in COMPS \setminus \Delta$), and hence, Δ is a current fluent diagnosis for Sys'. □

Lemma 3 If Δ is a current fluent diagnosis for Sys' then

$$SD \cup OBS \cup \{ab(c) \mid c \in \Delta\} \cup \{\neg ab(c) \mid c \in COMPS \setminus \Delta\}$$

is satisfiable.

Proof. Let Δ be a current fluent diagnosis for Sys'. Thus, $N = (SD', OBS' \cup OK_0)$ is consistent[11]. That means, there exists a model M of $(SD', OBS' \cup OK_0)$ such that $\Delta = \{c \,|M \models ab(c) \;\; \textbf{at} \;\; \mathsf{s}_C\}$. Let $M = (\Psi, \Sigma)$. From $OBS' \cup OK_0$ it is clear that $\Sigma(\mathsf{s}_C) = \Sigma(\mathsf{s}_1)$. Let $s = \Psi(\Sigma(\mathsf{s}_1))$. Since, s must be closed under the static causal laws of SD', which basically is SD, s must satisfy SD. Based on the construction of OBS', s must satisfy OBS. It then follows from the relation between M and Δ that s is a model of $SD \cup OBS \cup \{ab(c) \mid c \in \Delta\} \cup \{\neg ab(c) \mid c \in COMPS \setminus \Delta\}$, which proves the lemma. □

Lemma 4 Let $Sys = (SD, COMPS, OBS)$ be a system description and Sys' be the dynamic version of Sys. Assume that Δ is a diagnosis for Sys. Then, there exists two states s_1 and s_2 of Sys' such that

(a) $\{\neg ab(c) \mid c \in COMPS\} \subseteq s_1$;

(b) $OBS \cup \{ab(c) \mid c \in \Delta\} \cup \{\neg ab(c) \mid c \in COMPS \setminus \Delta\} \subseteq s_2$; and

(c) $s_2 = Cn(OBS \cup (s_1 \cap s_2) \cup \tilde{SD})$

Proof. (a) holds because of the assumption that $SD \cup \{\neg ab(c) \mid c \in COMPS\}$ is consistent. Let s_1 be a state satisfying (a).

For the easy of presentation of the proof, let $AB_{\neg\Delta} = \{\neg ab(c) \mid c \in COMPS \setminus \Delta\}$ and $AB_{\Delta} = \{ab(c) \mid c \in \Delta\}$. We define a sequence of set of literals of Sys', $\langle s_i \rangle_{i=0,\infty}$ as follows.

1. $s_0 = Cn(OBS \cup AB_{\Delta} \cup AB_{\neg\Delta} \cup \tilde{SD})$; and
2. for $n \geq 0$, $s_{n+1} = Cn(s_n \cup (s_n \cap s_1) \cup \tilde{SD})$.

The existence of s_0 is ensured by the assumption that Δ is a diagnosis of Sys while the existence of s_i $(i > 0)$ is guaranteed by the fact that Sys is consistent.

It is easy to see that $\langle s_i \rangle_{i=0,\infty}$ is a monotonic sequence (w.r.t set inclusion) and for every $i \geq 0$,

(*) s_i satisfies $\tilde{SD} \cup OBS \cup \{ab(c) \mid c \in \Delta\} \cup \{\neg ab(c) \mid c \in COMPS \setminus \Delta\}$.

Thus, due to the finiteness of the domain, we conclude that there exists an integer k such that $s_{k+1} = s_k$.

We prove that there exists a state s_2 such that

(**) $s_k \subseteq s_2$ and $s_2 = Cn(s_k \cup (s_2 \cap s_1) \cup \tilde{SD})$.

Obviously, if s_k is a complete set of literals w.r.t. the set of fluents of Sys', then s_k is such a state. Consider the case that s_k is incomplete. Then, the set $L = \{l \mid l \in s_1 \setminus s_k \text{ and } \neg l \notin s_k\}$ is not empty. Let $L_1 \subseteq L$ be the maximal set of literals such that $\tilde{SD} \cup s_k \cup L_1$ is consistent. So, we have that $\tilde{SD} \cup s_k \cup L_1 \models \{\neg l \mid l \in L \setminus L_1\}$. Therefore, $s_k \cup L_1 \models \{\neg l \mid l \in L \setminus L_1\} = Cn(s_k \cup L_1 \cup \tilde{SD})$. Thus, $s_2 = s_k \cup L_1 \cup \{\neg l \mid l \in L \setminus L_1\}$ satisfies (**).

We now prove that s_2 is a state satisfying (b) and (c). Because of (*) and $s_k \subseteq s_2$, we have that s_2 satisfies (b). We need to show that $s_2 = Cn(OBS \cup (s_2 \cap s_1) \cup \tilde{SD})$.

Let $S = Cn(OBS \cup (s_2 \cap s_1) \cup \tilde{SD})$. Consider a literal $l \in s_2$, we have the following cases:

[11] Due to Lemma 2, every diagnostic model of Sys' is a model of N.

1. $l \in s_k$, Since $AB_{\neg\Delta} \subseteq s_2 \cap s_1$, we have that $AB_{\neg\Delta} \subseteq S$. Furthermore, $OBS \subseteq S$. Thus, $OBS \cup AB_{\neg\Delta} \subseteq S$. By Proposition 3.3 of [Rei87], we have that $S \models AB_{\Delta}$. Hence, $OBS \cup AB_{\neg\Delta} \cup AB_{\Delta} \subseteq S$, i.e., $s_0 \subseteq S$. Using induction over i, we can prove that $s_i \subseteq S$ $(0 \leq i \leq k)$, and therefore, $s_k \subseteq S$. In other words, $l \in S$.

2. $l \in s_2 \setminus s_k$. Then, there are two sub-cases:

   (i) $l \in L_1$. Then, $l \in s_2 \cap s_1$. Hence, $l \in S$.

   (ii) $l \notin L_1$. By the construction of s_2, l has the form $\neg l'$ for some $l' \in L \setminus L_1$. Since $\tilde{SD} \cup s_k \cup L_1 \models \{\neg l' \mid l' \in L \setminus L_1\}$, we have that $l \in S$ too.

The above two cases show that $s_2 \subseteq S$.

Since s_2 and S are complete w.r.t. the set of fluents of Sys', we have that $s_2 = S$. Thus, s_1 and s_2 are two states of Sys' satisfying the lemma. □

References

[Bar95] C. Baral. Reasoning about Actions : Non-deterministic effects, Constraints and Qualification. In *Proc. of IJCAI 95*, pages 2017–2023, 1995.

[BCTD98] V. Brusoni, L. Console, P. Terenziani, and D. Dupre. A spectrum of definitions for temporal model based diagnosis. *Artificial Intelligence*, 102:39–79, 1998.

[BGP97] C. Baral, M. Gelfond, and A. Provetti. Representing Actions: Laws, Observations and Hypothesis. *Journal of Logic Programming*, 31(1-3):201–243, May 1997.

[BGP98] C. Baral, A. Gabaldon, and A. Provetti. Formalizing narratives using nested circumscription. *Artificial Inteligence*, 104(1-2):107–164, 1998.

[BLPZ99] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella. Diagnosis of large active systems. *Artificial Intelligence*, 110(1):135–183, 1999.

[BS98] C. Baral and T. Son. Formalizing sensing actions: a transition function based approach. In *Proc. of AAAI 98 Fall symposium on Cognitive robotics, (extended version at http://cs.utep.edu/chitta/chitta.html)*, 1998.

[CT94] M. Cordier and S. Thiébaux. Event-based diagnosis for evolutive systems. Technical report, Technical Report 819, IRISA, Cedex, France, 1994.

[dKMR92] J. de Kleer, A.K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.

[DL99] G. DeGiacomo and H. Levesque. Projection using regression and sensors. In *Proc. of IJCAI 99*, pages 160–165, 1999.

[FN92] G. Friedrich and W. Nejdl. Choosing observations and actions in model-based diagnosis/repair systems. In *Proc. of KR-92*, pages 489–498, 1992.

[HBR94] D. Heckerman, J. Breese, and K. Rommelse. Troubleshooting under uncertainty. Technical report, Microsoft Research, Technical Report MSR-TR-94-07, 1994.

[Lev96] H. Levesque. What is planning in the presence of sensing? In *AAAI 96*, pages 1139–1146, 1996.

[LR94] F. Lin and R. Reiter. State constraints revisited. *Journal of Logic and Computation*, 4(5):655–678, October 1994.

[McI97a] S. McIllraith. Representing actions and state constraints in model-based diagnosis. In *Proc. of AAAI-97*, pages 43–49, 1997.

[McI97b] S. McIllraith. *Towards a formal account of diagnostic problem solving.* PhD thesis, University of Toronto, 1997.

[McI98] S. McIllraith. Explanatory diagnosis: Conjecturing actions to explain observations. In *Proc. of KR-98*, 1998.

[MH69] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, Edinburgh, 1969.

[MNPW98] N. Muscettola, P. Nayak, B. Pell, and B. Williams. The new millennium remote agent: To boldly go where no ai system has gobe before. *Artificial Intelligence*, 103:5–48, 1998.

[Moo85] R. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985.

[MP92] Z. Manna and A. Pnueli. *The temporal logic of reactive and concurrent systems: specification.* Springer Verlag, 1992.

[MR92] S. McIllraith and R. Reiter. On tests for hypothetical reasoning. In W. Hamscher, L. Console, and J. de Kleer, editors, *Readings in Model-Based Diagnosis*, pages 89–96, 1992.

[MS94] R. Miller and M. Shanahan. Narratives in the situation calculus. *Journal of Logic and Computation*, 4(5):513–530, October 1994.

[MT95] N. McCain and M. Turner. A causal theory of ramifications and qualifications. In *Proc. of IJCAI 95*, pages 1978–1984, 95.

[Pin94] J. Pinto. *Temporal Reasoning in the Situation Calculus.* PhD thesis, University of Toronto, Department of Computer Science,, February 1994.

[PR93] J. Pinto and R. Reiter. Temporal reasoning in logic programming: A case for the situation calculus. In *Proc. of the 10th Int'l Conf. on Logic Programming, Hungary*, pages 203–221, 1993.

[Rei87] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.

[Rei91] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[Rei96] R. Reiter. Natural actions, concurrency and continuous time in the situation calculus. In L. Aiello, J. Doyle, and S. Shapiro, editors, *KR 96*, pages 2–13, 1996.

[Rei98] R. Reiter. *Knowledge in action: logical foundation for describing and implementing dynamical systems*. To appear, 1998. manuscript.

[Rym93] R. Rymon. *Diagnostic reasoning and planning in exploratory-corrective domains*. PhD thesis, University of Pennsylvania, 1993.

[SD89] P. Struss and O. Dressler. Physical negation: Integrating fault models into the general diagnostic engine. In *Proc. of IJCAI 89*, pages 1318–1323, 1989.

[SL93] R. Scherl and H. Levesque. The frame problem and knowledge producing actions. In *AAAI 93*, pages 689–695, 1993.

[SSL+96] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, , and D. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Trans. on Control Systems Technology*, 4(2):105–124, 1996.

[Str97] P. Struss. Fundamentals of model-based diagnosis of dynamic systems. In *Proc. of IJCAI 97*, pages 480–485, 1997.

[SW93] Y. Sun and D. Weld. A framework for model-based repair. In *Proc. of AAAI 89*, pages 182–187, 1993.

[TCJK96] S. Thiébaux, M.O. Cordier, O. Jehl, and J.P. Krivine. Supply restoration in power distribution systems – a case study in integrating model-based diagnosis and repair planning. In *Proc. of UAI 96*, pages 525–532, 1996.

[Thi97] M. Thielscher. A theory of dynamic diagnosis. *ETAI*, 2(11), 1997.

[Tur97] H. Turner. Representing actions in logic programs and default theories. *Journal of Logic Programming*, 31(1-3):245–298, May 1997.

[WN96] B. Williams and P. Nayak. A model-based approach to reactive self-configuring systems. In AAAI 96, pages 971–978, 1996.

[WN97] B. Williams and P. Nayak. A reactive planner for a model-based executive. In *IJCAI 97*, 1997.