Diagnostic Problem Solving via Planning with Ontic and Epistemic Goals

Jorge A. Baier
Depto. de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Santiago, Chile

Brent Mombourquette
Department of Computer Science
University of Toronto
Toronto, Canada

Sheila A. McIlraith
Department of Computer Science
University of Toronto
Toronto, Canada

Abstract

Diagnostic problem solving involves a myriad of reasoning tasks associated with the determination of diagnoses, the generation and execution of tests to discriminate diagnoses, and the determination and execution of actions to alleviate symptoms and/or their root causes. Fundamental to diagnostic problem solving is the need to reason about action and change. In this work we explore these myriad of reasoning tasks through the lens of artificial intelligence (AI) automated planning. We characterize a diversity of reasoning tasks associated with diagnostic problem solving, prove properties of these characterizations, and define correspondences with established automated planning tasks and existing state-of-the-art planning systems. In doing so, we characterize a class of planning tasks with epistemic and ontic goals which we show can be compiled into non-epistemic planning, allowing state-of-the-art planners to compute plans for such tasks. Furthermore, we explore the effectiveness of using the conditional planner Contingent-FF with a number of diagnostic planning tasks.

1 Introduction

Automated diagnosis seeks to determine what is wrong with a system, prompted by some observations of egregious behaviour. As our world becomes increasingly instrumented with sensors – our street corners, our homes, our cars, and even our bodies – and as the infrastructure that controls our power, communication, and transportation systems grows in complexity, we must rely on computers to monitor the state of these systems and to oversee their operation. Unfortunately, these systems can and do malfunction, resulting in diagnostic problems of such enormous complexity that they confound human reasoning.

Diagnostic Problem Solving (DPS) refers to the myriad of reasoning tasks associated with the diagnosis, testing, and repair of a system. In this work we advocate for a *purposeful view of diagnostic problem solving*. While a naïve approach to DPS suggests that we generate candidate diagnoses, identify a unique diagnosis through testing, and then treat or repair, we instead observe that identifying candidate diagnoses may be unnecessary or perhaps only necessary to the extent that it informs an appropriate course of action to be selected – a course of action that may result in the realization of further tests to discriminate diagnoses, or to alleviate the symptoms or potential root causes, potentially without actually identifying a unique root cause.

Example 1 *Consider a run-of-the-mill flashlight that is not emitting light. A common response is to turn the flashlight on and off a few times. If it's still malfunctioning, the most likely hypothesis is that the batteries are dead, but it could also be the case that the bulb is burned out, or that there is a loose connection somewhere. That's three candidate diagnoses, and there could be more. A typical course of action would be to open up the flashlight, take out the batteries, put in new ones, re-assemble the flashlight and turn it on again. If the flashlight emits light, you'll likely be happy, recycle the batteries and consider yourself "done." Your purpose was* not *to diagnose the flashlight, but rather to get it working again. A more careful examination of what went on shows that the course of action you took, together with the happy outcome that the flashlight is now "working," served to eliminate the hypothesis that the bulb was burned out, and it tightened the connection, effectively* repairing *the connection regardless of whether it was faulty or not. This changed the space of hypotheses under consideration. What is equally interesting is that this sequence of actions neither confirmed nor refuted the hypothesis that the batteries were dead. The cause of the faulty behaviour could have been the result of a loose connection which got fixed in the process of changing the batteries. Those batteries you recycled could still be OK! What's also noteworthy is that if after executing the procedure the flashlight had not emitted light, you would still be left with the hypotheses that the original batteries were dead or that the bulb was broken, but you would also have the further (granted, unlikely) hypothesis that the new batteries were also dead.*

The above seemingly simple DPS scenario illustrates the need for reasoning about action and change as well as reasoning about knowledge, and in particular what an agent comes to know about aspects of the world (symptoms and diagnoses in this case) based upon the execution of both world-altering and sensing actions. It is also suggestive of the somewhat subordinate role the actual candidate diagnoses may play in the resolution of a system failure scenario.

In this paper, we explore this purposeful view of diagnostic problem solving through the lens of AI automated planning. Our motivation for doing so is pragmatic. We are interested in characterizing the fundamental knowledge representation and reasoning tasks that underlie DPS in its many guises, but we wish to do so in a manner that supports establishing its correspondence with the state of the art in AI automated planning theory and practice. There have been tremendous advances in AI automated planning systems over the last decade and a number of highly optimized planning systems exists. Further there have been significant recent advances in non-classical planning such as conformant or contingent planning that support planning with incomplete information and/or with sensing. A major contribution of this paper is to show that many purposeful DPS tasks that have been heretofore unsolvable can be realized in varying measure through recent advances in state-of-the-art AI planning systems. Our characterization not only allows us to solve certain problems now, but it provides the insight and understanding that will support the realization of these and isomorphic tasks as computational machinery in planning improves in the coming years.

Characterizing diagnostic problem solving tasks in terms of planning builds upon a large body of research that includes research on topics as varied as reasoning about knowledge and action (e.g., Scherl and Levesque 2003; Petrick and Bacchus 2002), planning with epistemic goals (e.g., Herzig, Lang, and Marquis 2003; Aucher and Bolander 2013), and conformant (e.g., Palacios and Geffner 2009), contingent (e.g., To, Pontelli, and Son 2011), and classical (e.g., Helmert 2006) planning. We discuss this related research later in the paper.

In Section 2 we introduce the mathematical formalisms upon which our work rests. In Section 3 we introduce the notion of a diagnosis that we use here, and contrast it to other forms of diagnosis discussed in the literature. With a definition of diagnosis in hand, in Section 4 we present the notion of a diagnostic plan, establish its correspondence to known planning paradigms, and establish properties of various forms of diagnostic plans. In Section 5 we introduce a notion of epistemic diagnostic planning – planning to determine a particular diagnosis, or to discriminate diagnoses. We define compelling classes of epistemic goals and then show that it is possible to use state-of-the-art non-epistemic planners to plan for such epistemic goals, by providing a sound and complete translation of such epistemic tasks to conditional planning. In Section 6 we turn our attention to the realization of our newly established diagnostic planning tasks via existing planning systems. We conclude with some reflections on our work, its relationship to other work and prospects for future work.

2 Preliminaries

In this section we introduce a planning language that will allow us to define various kinds of planning tasks which we then will show can model interesting DPS tasks. In particular, we define a common language for deterministic (classical), conformant, and conditional planning.[1] The planning language we present below builds on the ADL planning language (Pednault 1989), and considers extensions for uncertainty about the initial state and conditional plans that have been presented in a similar way by other researchers (e.g., Palacios and Geffner 2006; To, Pontelli, and Son 2011).

2.1 Dynamical Systems

Dynamical systems can be formally described in many ways. In this paper we model them as transition systems, which we represent with a standard planning language. As such, transitions occur as the result of actions described in terms of preconditions and effects, and the domain is finite (i.e., there is a finite number of system configurations). Formally, a dynamical system is a tuple $\Sigma = (F, A, \Omega, I)$, where F is a finite set of fluent symbols, A is a set of deterministic actions, Ω is a set of sensing actions, and I is a set of clauses over F that defines a set of possible initial states. If $p \in F$, then p and $\neg p$ are *fluent literals*. If ℓ is a literal, we denote its complement by $\overline{\ell}$; thus, $\overline{p} = \neg p$ and $\overline{\neg p} = p$. Every action $a \in A$ is defined by a precondition $prec(a)$, which is a conjunction of fluent literals, and $\mathit{eff}(a)$, a set of conditional effects of the form $C \to L$, where C is a conjunction of fluent literals and L is a fluent literal. We sometimes write the unconditional effect $\to L$ as simply L, and use $true$ to denote an empty precondition. Each sensing action, on the other hand, is defined by its precondition $prec(a)$, which is a conjunction of fluent literals, and $obs(a)$, which is the fluent literal that is observed by the sensing action.

A *system state* s is a set of fluent symbols, which intuitively defines all that is true in a particular state of the dynamical system. For a system state s, we define $M_s : F \to \{true, false\}$ as the truth assignment that assigns the truth value $true$ to p if $p \in s$, and assigns $false$ to p otherwise. We say a state s is *consistent* with a set of clauses $\mathcal{C}$, if $M_s \models c$, for every $c \in \mathcal{C}$.

We denote by S_0 the set of planning states consistent with the clauses of the initial state I. We say a dynamical system has a *complete initial state* iff $|S_0| = 1$; i.e., I has only one model. Σ has an *incomplete initial state* iff $|S_0| > 1$. We formalize the notion of conditional plans through *action trees* as follows.

Definition 1 (Action Tree) *Given a system $\Sigma = (F, A, \Omega, I)$, an action tree T is:*

- ϵ *(the empty tree); or*
- aT'*, where $a \in A$, and T' is an action tree; or*
- $a(T', T'')$*, where $a \in \Omega$ and T' and T'' are action trees.*

We denote by $\mathcal{T}_\Sigma$ the set of action trees in Σ. Furthermore, we say that an action a is *executable* in a state s if $M_s \models$

[1] A paradigm related to conditional planning is *contingent planning*. Although originally understood to define the same class of problems as conditional planning, current research in contingent planning frequently sees contingent planning as an incremental process in which planning is intertwined with execution (see e.g., Brafman and Shani 2012). We thus stick here to conditional planning to emphasize that we are looking for a conditional plan in an offline manner.

$prec(a)$. If $a \in A$ is executable in a state s, we define its successor state as $\delta(a, s) = (s \setminus Del) \cup Add$, where Add contains a fluent f iff $C \rightarrow f$ is an effect of a and $M_s \models C$. On the other hand Del contains a fluent f iff $C \rightarrow \neg f$ is an effect of a, and $M_s \models C$.

Now we define how to compute the set of states that result from executing an action tree. To that end, we define the relation $\vdash_\Sigma$: $\mathcal{T}_\Sigma \times 2^F$ such that $(T, S) \vdash_\Sigma (T', S')$ intuitively means that if the agent is in any of the states in S, then performing one step of T results in being in some state in S', with action tree T' remaining. Formally,

- $(aT, S) \vdash_\Sigma (T, S')$ if a is executable in every state in S, and $S' = \{\delta(a, s) \mid s \in S\}$.
- $(a(T', T), S) \vdash_\Sigma (T, S')$ if a is executable in every state in S and $S' = \{s \in S \mid M_s \models obs(a)\}$ or $S' = \{s \in S \mid M_s \not\models obs(a)\}$.

We use $\vdash$ instead of $\vdash_\Sigma$ when the transition system is obvious from the context. We denote by $\vdash^*$ the reflexive and transitive closure of $\vdash$. This allows us to define what states result from executing an action tree in the initial state of a transition system.

Definition 2 (Resulting State) *Given a transition system Σ, state s is a resulting state from executing action tree T in Σ iff $(T, S_0) \vdash^* (\epsilon, S')$ and $s \in S'$.*

2.2 Classical, Conformant, and Conditional Planning

Below we define classes of planning tasks that have been studied extensively by the planning community. Deterministic (classical) planning is the most standard form of planning in which there is a unique initial state, and hence knowledge about the state of the world is always complete. In conformant planning, on the other hand, there is uncertainty but no observability, and actions that change the state of the world are deterministic; hence solutions to these tasks are sequences of actions. Finally, in conditional planning,[2] there is uncertainty about the initial state and the agent may observe the world through sensing actions. The resulting plan in this case is typically an action tree.

Definition 3 (Classes of Planning Tasks) *Given a set of literals G, and a system $\Sigma = (F, A, \Omega, I)$ we define the following classes of planning tasks:*

- (Σ, G) *is a* deterministic *or* classical *planning task if I defines a complete initial state and $\Omega = \emptyset$.*
- (Σ, G) *is a* conformant *planning task if I does not define a complete initial state and $\Omega = \emptyset$.*
- (Σ, G) *is a* conditional *planning task if I does not define a complete initial state and $\Omega \neq \emptyset$.*

Definition 4 (Plan) *Action tree T is a plan for planning task (Σ, G) iff for every state s_f that may result from the execution of T in Σ it holds that $M_{s_f} \models G$.*

When T is a plan for a deterministic task, then we say T is a *deterministic plan*. Analogously, we use the terms *classical plan*, *conformant plan*, and *conditional plan*.

3 Characterizing Diagnoses

Automated diagnosis has long been a problem of interest to the AI community. Well-publicized early work focused on expert systems approaches as exploited in the medical diagnosis expert systems MYCIN (Shortliffe and Buchanan 1975). In the mid-1980's AI researchers turned their attention to model-based diagnosis. Early work in this area included Geneserth's DART system (Genesereth 1984), as well as GDE, the General Diagnosis Engine by de Kleer and Williams (de Kleer and Williams 1987) among others. In 1987 Reiter published his seminal paper formalizing the notion of *consistency-based diagnosis* and *minimal diagnosis*, which were predicated on a so-called first principles model of the normative behaviour of a system (Reiter 1987). This characterization defined a diagnosis to be a minimal set of components that must be designated as *abnormal* in order for observations of system behaviour to be logically consistent with the model (axiomatization) of the system. The exploitation of fault models and other aspects of system behaviour and function necessitated a more stringent characterization of diagnosis in terms of *abduction* (e.g., (Console and Torasso 2006), (Poole 1994)) in which faults or other explanations were posited in order to *entail* or otherwise explain observations. Such abductive diagnoses were originally conceived to work with fault models, rather than normative models, in order to entail faulty behaviour. de Kleer, Mackworth, and Reiter published a follow-on to Reiter's 1987 paper in 1992 that combined aspects of abductive and consistency based diagnosis into kernel diagnoses (de Kleer, Mackworth, and Reiter 1992). All of these characterizations of diagnosis related to static systems, described using a triple $(SD, COMPS, OBS)$ – the system description, a finite set of components to be diagnosed, and an observation. There was no notion of system dynamics – just a single state.

The systems we are concerned with in this paper include not only these static descriptions but also a rich theory of action that supports diagnostic planning in its many guises. We begin with a definition of such a diagnostic system.

- SD, the system description, a set of propositional sentences;
- $COMPS$, the components, a finite set of constants;
- OBS, the observation, a conjunction of ground literals:
- $\Sigma = (F, A, \Omega, I)$, a dynamical system that describes actions relevant to diagnostic problem solving tasks associated with the system described by SD.

We illustrate these with a simple example.

Example. Consider a flashlight, comprised of a battery and a switch. If the switch is on and both the battery and switch are operating normally, then light will be emitted. This fact can be described by the following logical formula, which is included in the system's description SD:

$$on \wedge \neg AB(battery) \wedge \neg AB(switch) \supset light.$$

[2] Early literature in conditional planning (e.g. Pryor and Collins 1996) assumed complete observability of the world. However, the current standard in the planning community—under which state-of-the-art solvers are developed—is to consider sensing actions as the only mechanism to observe the world (e.g. To, Pontelli, and Son 2011).

a	$prec(a)$	Effect/Observation
$turn\text{-}on$	$\neg on$	on
$change\text{-}battery$	$true$	$\neg AB(battery)$
$fix\text{-}switch$	$true$	$\neg AB(switch)$
$sense\text{-}light$	$true$	$light$

Table 1: Preconditions and direct effects of the actions in our flashlight example.

The set of system components is simply defined by $COMPS = \{battery, switch\}$.

Now we assume the flashlight can be operated by a human user. The available actions are: turn on the switch, change the battery, and fix the switch. In addition, we have the ability to observe whether or not there is light in the room. The dynamics of these actions is described using a transition system Σ, where $F = \{on, AB(switch), AB(battery), light\}$, $A = \{turn\text{-}on, change\text{-}battery, fix\text{-}switch\}$, and $\Omega = \{sense\text{-}light\}$. The effects of the actions are described using our planning language as shown by Table 1. §

In general, the action theory described by Σ supports reasoning about actions to test, repair, or eradicate egregious behaviour in a system. On the other hand, the system description describes complex interactions between the components of the system.

Our notion of *diagnostic system*, which we define formally below, intuitively integrates the system description SD with the dynamics of the world described by Σ. In doing so, we need to address the so-called *ramification problem* – the problem of characterizing the indirect effects of actions. To see why this is, in our example, observe that performing the action $turn\text{-}on$ may have the indirect effect of $light$, under the condition $\neg AB(battery) \wedge \neg AB(switch)$.

The ramification problem is a well-studied problem in the KR community (e.g., Lin 1995; McCain and Turner 1995; Thielscher 1995; Sandewall 1996; Pinto 1999; Strass and Thielscher 2013) and has also been previously studied in the specific context of DPS (e.g., McIlraith 1997; McIlraith 2000; McIlraith and Scherl 2000). For the purposes of this paper, we adopt an existing solution to the ramification problem that compiles the indirect effects of actions into additional direct effects of actions proposed both by Pinto (1999) and in variation most recently by Strass and Thielscher (2013). The solution is predicated on augmentation of the constraints, SD, that additionally captures the causal relationship between fluents. These causal relationships have historically been captured via an explicit causal relation in the constraints (e.g., Lin 1995; McCain and Turner 1995) or by the augmentation of SD with a causal graph structure that ranges over the literals in SD (e.g., McIlraith 2000).

Example (continued). In our flashlight example, we consider the following ramification constraint, which is extracted directly from SD.

$$on \wedge \neg AB(battery) \wedge \neg AB(switch) \ \textbf{causes} \ light.$$

a	New Effect
$turn\text{-}on$	$\neg AB(battery) \wedge \neg AB(switch) \rightarrow light$
$change\text{-}battery$	$on \wedge \neg AB(switch) \rightarrow light$
$fix\text{-}switch$	$on \wedge \neg AB(battery) \rightarrow light$

Table 2: Additional effects of actions in our example theory.

From there, we use Pinto's algorithm (1999) to compute additional effects for the actions in our theory. Some of the resulting effects are shown in Table 2. §

Now we provide a formal definition for a diagnostic system.

Definition 5 (Diagnostic System) *Given SD, $COMPS$, OBS, and $\Sigma = (F, A, \Omega, I)$, as defined above, a diagnostic system Σ_{SD} is a tuple (F', A', Ω, I') where:*

- *F' contains the elements in F and in $Vars(SD)$, the propositional variables in SD, and ground fluents of the form $AB(c)$ for every $c \in COMPS$;*
- *A' contain the same actions in A but augmented, following Pinto (1999), with conditional effects to address the ramification problem that emerges from the integration of Σ and SD;*
- *$I' = I \cup SD \cup OBS$.*

Note that the initial state I' contains both the system's description and the observation. In our example, we could have that $I = \{on\}$ and that $OBS = \{\neg light\}$. By including SD in I' we enforce that states consistent with I' have at least one abnormal component. Note that an alternative means of characterizing a diagnostic system is to treat ramifications as further actions that are triggered by the direct effects of actions.

We now formally define the notion of diagnosis, which we borrow from de Kleer, Mackworth, and Reiter (1992)'s, in which we posit a minimal subset of components that must be behaving abnormally in order to account for the observation in the initial state of our dynamical system. To facilitate explication, we appeal to a characterization of diagnosis in terms of abnormal components; however the work in this paper is applicable to a diversity of definitions of diagnoses or hypotheses and need not rely on the use of distinguished components. The generalization is straightforward. Recall that I', of Σ_{SD}, includes our observation, OBS of (potentially egregious) behaviour.

Definition 6 (Diagnosis) *Given a diagnostic system Σ_{SD}, $\Delta \subseteq COMPS$ is a diagnosis iff*

$$I' \cup \bigcup_{c \in \Delta} AB(c) \cup \bigcup_{c' \in COMPS \setminus \Delta} \neg AB(c')$$

is satisfiable.

Definition 7 (Minimal Diagnosis) *Δ is a minimal diagnosis of Σ_{SD} if Δ is a diagnosis and no other proper subset Δ' of Δ is a diagnosis.*

In previous work, we examined diagnosis of dynamical systems with respect to a narrative of observations that evolved over a period of time. We characterized the notion of an *explanatory diagnosis* in the situation calculus (Sohrabi, Baier, and McIlraith 2010; McIlraith 1998) and relatedly the notion of an *explanation* in a planning-inspired propositional language (Sohrabi, Baier, and McIlraith 2011). These definitions of diagnosis and explanation conjecture a set of zero or more assumptions together with a sequence of actions to account for the observations. These definitions of diagnosis and explanation are not at odds with what is proposed here. Indeed, a diagnosis, Δ would hold in the final state of an explanatory diagnosis, with OBS as the final observation of the narrative of observations used to construct the explanatory diagnosis. The generation of explanatory diagnosis looks back in time to conjecture what happened based on past observations. Observations going forward are integrated with the diagnostic plans.

4 Diagnostic Planning

While automated diagnosis remains a well-studied area of research, we argue for a purposeful view of diagnosis. In particular, rather than generating candidate diagnoses and performing tests to identify a unique diagnosis, after which a course of action (amelioration/treatment) is determined, we argue that the determination of a unique diagnosis is generally not an end in itself and that a pragmatic view of DPS should focus on acting.

One argument in support of this viewpoint is that in many settings there are a limited number of courses of action that can achieve a particular goal. These actions/plans induce an equivalence class of diagnoses for which a particular course of action is relevant. For example, many families of bacterial infection require treatment with the same antibiotic and we need not completely discriminate the nature and location of the infection before treating. Similarly, if a photocopier is malfunctioning, despite the ability to perform an in-depth diagnosis, the first course of action is often to turn the machine off and on, resolving a large family of faults simultaneously without the need for time-consuming, and time-wasting in this case, differential diagnosis.

All of the diagnostic planning tasks we examine take the same general form. The dynamical model of the system, Σ_{SD}, is augmented with additional information about the initial state, the goal state and some constraints that need to be enforced throughout execution of the plan. In this section, we provide a general formulation of a diagnostic plan. We discuss the diversity of diagnostic tasks that can be achieved via the specification of different initial conditions, goals, and constraints, and the complexity of plan existence in some of these different settings. While the types of plans we are interested in here change the state of the world, we are also interested in plans that are designed to change our state of knowledge without necessarily changing (much of) the world. In the section that follows, we look at epistemic goals – planning to know whether or not a diagnosis is true, to refute a diagnosis, or to discriminate a collection of candidate diagnoses.

Definition 8 (Diagnostic Plan) *Given a diagnostic planning task* $(\Sigma_{SD}, Init, \Phi, G)$ *where*

- $\Sigma_{SD} = (F, A, \Omega, I)$*, is the diagnostic system;*
- $Init$*, is a set of logical formulae that provide additional information about the initial state;*
- Φ*, is a logical formula representing additional state constraints that must be enforced throughout the plan; and*
- G*, is a set of literals that prescribe the diagnostic planning goal,*

and where $Init \cup I \cup \Phi$ *is satisfiable.*

Action tree T *is a diagnostic plan for the diagnostic planning task* $((F, A, \Omega, I \cup Init), G)$ *under the constraint of* Φ *iff* T *is a plan for the planning task* $((F, A, \Omega, I \cup Init), G)$*, and for every* S *such that* $(T, S_0) \vdash^* (T', S)$ *it holds that* $M_s \models \Phi$*, for every* $s \in S$*.*

4.1 Properties of Diagnostic Plans

As evident from the definition presented, our characterization of a diagnostic plan is coupled to our previously defined classes of planning tasks. These classes differ with respect to the completeness of their initial states and whether they exploit sensing of any form. Such characteristics have implications with respect to the complexity of planning.

The following are the complexity classes of the decision problem for the different planning tasks.

Theorem 1 (Diagnostic Planning w/ Complete Info.) *Given a diagnostic planning problem* $(\Sigma_{SD}, Init, \Phi, G)$*, if* $I \cup Init \cup \Phi$ *defines a complete initial state and if* $\Omega = \emptyset$ *then this is a* classical *planning task and deciding whether there exists such a plan is PSPACE-complete.*

This result follows from Bylander's result on the complexity of deterministic planning (Bylander 1994).

Theorem 2 (Diagnostic Planning without Sensing) *Given a diagnostic planning problem* $(\Sigma_{SD}, Init, \Phi, G)$*, if* $\Omega = \emptyset$ *but* $I \cup Init \cup \Phi$ *does not define a complete initial state then this is a* conformant *planning problem and deciding whether there exists such a plan is EXPSPACE-complete.*

This follows from Haslum's result on the complexity of conformant planning (Haslum and Jonsson 1999). Finally,

Theorem 3 (Diagnostic Planning with Sensing) *Given a diagnostic planning problem* $(\Sigma_{SD}, Init, \Phi, G)$*, if* $I \cup Init \cup \Phi$ *does not define a complete initial state and* $\Omega \neq \emptyset$ *then this is a* conditional *planning problem and deciding whether there exists such a plan is 2-EXPTIME-complete.*

This follows from Rintanen's results on the complexity of conditional planning (Rintanen 2004).

4.2 The Many Guises of Diagnostic Planning

Our characterization of a diagnostic plan encompasses a diversity of DPS scenarios. Here we informally explore some of these varied scenarios in order to illustrate the broad utility of our characterization. Each scenario is realized by varying the $Init$ and G and can be performed with or without sensing as the scenario necessitates, and with commensurate implications regarding the type of planner.

Eradicate egregious behaviour By setting $Init = \emptyset$ and $G = \neg OBS$ we can plan to eradicate behaviour without first explicitly computing a diagnosis. If sensing is permitted, sufficient information will be garnered to select an appropriate course of action. Such information may or may not entail a unique diagnosis as illustrated in some of the previous examples. Without sensing, a conformant plan will be generated (where possible) that will work regardless of what is wrong with the system.

Fix the system, given a diagnosis Given a diagnosis Δ, $Init = \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in COMPS \setminus \Delta\}$ and $G = \bigwedge_{c \in \Delta} \neg AB(c)$. When a unique diagnosis has been determined, this can be added to the initial state and a plan generated accordingly. In such a case I may transform or be transformable into a classical planning problem, greatly diminishing the computational effort involved in generating a plan. $Init$ may also be used to capture the set of candidate diagnoses. (E.g., *The car won't start because it's either out of gas or the battery is dead.*) Even in a scenario such as this one, the best plan may require no sensing since this candidate diagnosis set may dictate the same "fix" – call for road-side assistance.

Assume a diagnosis and fix it or eradicate behaviour We can also use the planner to do what-if planning. Whereas the previous scenario used $Init$ to add facts to the initial state, it is often compelling to *assume* a particular likely diagnosis and generate a plan that will work predicated on the assumption being correct. (E.g., *Assume the battery is dead and fix it.*) A subset of candidate diagnoses may similarly be assumed.

Discriminate between different diagnoses Given two diagnoses Δ_1 and Δ_2, return a plan to determine which one of Δ_1 or Δ_2 may be causing the egregious behaviour. We require a notion of *epistemic goals*, which we define below, to formalize this as a planning problem.

Our definition of diagnostic plan also supports the enforcement of (safety) constraints. There are a number of compelling uses for such constraints, particularly in cases where a diagnosis is being assumed. In such a scenario, the user may wish to prescribe avoidance of things that would be fatal should the assumptions be flawed (e.g., giving penicillin to a patient with allergies). Constraints of this form can often be compiled away so as not to add to the difficulty of planning. Finally, the user can exercise flexibility in eliminating sensing actions to generate conformant rather than contingent plans, or to enforce Φ by eliminating the actions that would result in its violation.

5 Epistemic Diagnostic Planning

In the previous section we examined tasks to achieve some state of the world. Here we wish to generate plans to achieve epistemic goals. For example, we may wish to generate a plan to *know* a particular diagnosis (*E.g., I know Ralph has meningitis.*), to discriminate between diagnoses (*E.g., I know Ralph has one and only one of meningitis, strep-A, or influenza.*), or to eliminate a diagnosis (*E.g, I've eliminated the possibility that Ralph has strep-A.*).

The notion of planning and reasoning to achieve a state of knowledge dates back to work by Moore (1985). Scherl and Levesque (2003) later integrated Moore's approach with Reiter's solution to the frame problem (1991). More recently, epistemic planning has been discussed in the context of dynamic epistemic logic (e.g, (Herzig, Lang, and Marquis 2003; Andersen, Bolander, and Jensen 2012).

Currently, there are no competitive planning systems that implement the possible world semantics. There are a few systems that however implement the idea of *knowledge-level planning* (Demolombe and Pozos Parra 2000; Petrick and Bacchus 2002), in which the knowledge of the agent is explicitly represented as propositions of the language (e.g., using a fluent KF to represent that the agent knows F). These planners are capable of carrying out a limited, but still reasonably expressive form of reasoning about knowledge. A few systems like PKS (Petrick and Bacchus 2002) and the web-service composition planner by Pistore et al. (2005) adopt the approach of *planning at the knowledge level* to implement these ideas. Planning at the knowledge level may achieve good performance in many planning benchmarks however, their ability to reason about knowledge is limited. Along the same lines, Palacios and Geffner (2009) proposed a compilation technique that maps contingent planning into knowledge-level deterministic tasks. As expected, their translation is compact, sound, and complete for only a certain class of problems (Palacios and Geffner 2009).

While many simple epistemic diagnostic planning tasks can indeed be mapped into knowledge-level planning domains we propose an alternative compilation into conditional planning with sensing actions. Our motivation is practical since this enables computing diagnoses with a variety of existing planning systems.

Our translation still takes a somewhat impoverished view of the world, choosing not to appeal to rich modal theories of knowledge and belief in favour of adopting the stance of planning at the so-called *belief level*. In this view, the state of the system captures the agent's beliefs about the world. Specifically, in the initial state, we assume that the agent knows any formula ϕ that is such that for every s consistent with I it holds that $M_s \models \phi$. Similarly, when the agent performs actions, the agent knows all formulae that hold in all states that it could reach. Formally, given a set of states S, and a formula ϕ, we say that:

$$K(\phi, S) \text{ iff } M_s \models \phi \text{ for each } s \in S,$$

where the intuitive meaning of $K(\phi, S)$ is that the agent knows ϕ when the set of states it is possibly in is S.

With this definition in hand we are ready to define our notion of planning with epistemic goals.

Definition 9 (Epistemic Plan) *Let $\Sigma = (F, A, \Omega, I)$ be a transition system and S_0 be the set of all plan states consistent with I. Then,*

- *T is a plan for $\mathsf{Know}(\phi)$ iff for every S such that $(T, S_0) \vdash^* (\epsilon, S)$ it holds that $K(\phi, S)$,*
- *T is a plan for $\mathsf{KnowWhether}(\phi)$ iff for every S such that $(T, S_0) \vdash^* (\epsilon, S)$ it holds either $K(\phi, S)$ or $K(\neg\phi, S)$, and*
- *T is a plan for $\mathsf{Discriminate}(\phi, \psi)$ iff for every S such that $(T, S_0) \vdash^* (\epsilon, S)$ it holds either $K(\phi \wedge \neg\psi, S)$ or $K(\neg\phi \wedge \psi, S)$.*

Under this definition it is simple to prove that T is a plan for ϕ if and only if T it is an epistemic plan for $\mathsf{Know}(\phi)$. This is compatible with our notion of planning at the belief level: the knowledge of the agent is captured by the set of states the agent is at. This also means that if one wants a plan for $\mathsf{Know}(\phi)$, then one can obtain such a plan by querying a regular conditional planner for the goal ϕ. However, it is not immediately straightforward how to obtain plans for the other types of epistemic goals using a conditional planner. For example, using the non-epistemic goal $\phi \vee \neg\phi$ instead of $\mathsf{KnowWhether}(\phi)$ would not work since $\phi \vee \neg\phi$ is tautological and therefore achieved by every action tree.

Interestingly, it *is* possible to treat all types of epistemic goals as ontic planning. Below we propose a simple compilation which maps our notion of planning with epistemic goals to ontic conditional planning.

5.1 From Epistemic Planning to Conditional Planning

Given an epistemic goal of the form $\mathsf{KnowWhether}(\phi)$, the main idea underlying this compilation is to add an additional fluent $kw\text{-}\phi$ that can be used to replace the epistemic goal $\mathsf{KnowWhether}(\phi)$.

To simplify the presentation we will assume, for now, that the goal is $\mathsf{KnowWhether}(L)$, where L is a literal, and that as usual $\Sigma = (F, A, \Omega, I)$ is the transition system. The compiled planning problem is $\Sigma' = (F', A', \Omega, I)$, and F' and A' are generated by performing the following steps:

1. Let F' be $F \cup \{kw\text{-}L\}$.
2. Add to A' all actions in A plus actions $kw\text{-}act\text{-}pos\text{-}L$ and $kw\text{-}act\text{-}neg\text{-}L$, whose preconditions are, respectively, L and $\overline{L}$. Both actions have a single effect, $kw\text{-}L$.
3. For each action a in A' that contains either $C \to L$ or $C \to \overline{L}$ as a conditional effect, for some C different from $\{true\}$, we add the unconditional effect $\neg kw\text{-}L$ to a.

Note that Step 2 generates actions that add the fact $kw\text{-}L$. These actions can only be performed if the set of states the agent believes it is in, say, S, is such that $K(L, S)$ or $K(\neg L, S)$ (i.e., the agent knows whether L is true). Step 3 handles the case in which there may be "loss of knowledge" due to a conditional effect of an action. To see this, imagine a system with a single action A which is always executable and which has conditional effect $p \to L$. Assume I is such that the set of states consistent with I is $\{\{p\}, \{\}\}$. Even though the agent knows whether L in the states of I, it is not the case anymore after performing A, since the set of resulting states is $\{\{p, L\}, \{\}\}$.

We now prove that our proposed translation is sound and complete, in the senses defined below.

Theorem 4 (Completeness) *Let $\Sigma = (F, A, \Omega, I)$ be a transition system, and let Σ' be defined as above. If T is a plan for $(\Sigma, \mathsf{KnowWhether}(L))$ then there exists a plan T' for $(\Sigma', kw\text{-}L)$ which differs from T only in that it contains actions of the form $kw\text{-}act\text{-}pos\text{-}L$ or $kw\text{-}act\text{-}neg\text{-}L$ at the end of each of T''s branches.*

Proof. We note that if there exists an action tree T that is a plan for $(\Sigma, \mathsf{KnowWhether}(L))$ then we can construct a plan T' for $(\Sigma', kw\text{-}L)$ by simply adding an additional $kw\text{-}act\text{-}pos\text{-}L$ or $kw\text{-}act\text{-}neg\text{-}L$ as the final action in each branch of the tree. Such actions are executable at that point since it holds that $K(L, S)$ or $K(\neg L, s)$, where S is the set of states reached by that branch. ■

Theorem 5 (Soundness) *Let $\Sigma = (F, A, \Omega, I)$ be a transition system, and let Σ' be defined as above. If T is a plan for $(\Sigma', kw\text{-}L)$ then, by removing all actions of the form $kw\text{-}act\text{-}pos\text{-}L$ or $kw\text{-}act\text{-}neg\text{-}L$ from T we obtain an action tree that is a plan for $(\Sigma, \mathsf{KnowWhether}(L))$.*

Proof. Let T be a plan for $kw\text{-}L$. Take any S_f that results from the execution of a branch of T; i.e., such that $(T, S_0) \vdash^* (\epsilon, S_f)$. Observe that $kw\text{-}L \in S_f$. Now let (kT', S) be the configuration visited by the execution of the branch in which kwL is added for the last time (here k is either $kw\text{-}act\text{-}pos\text{-}L$ or $kw\text{-}act\text{-}neg\text{-}L$). In other words, let S be such that $(T, S_0) \vdash^* (kT', S) \vdash (T_1, S_1) \vdash (T_2, S_2) \vdash^* (T_n, S_n)$, with $T_n = \epsilon$, $S_n = S_f$, and such that for all $i \in \{1, \ldots, n\}$, $kw\text{-}L \in S_i$. Because k is executable in S, either $K(L, S)$ or $K(\neg L, S)$ holds. Furthermore, because S and S_1 differ in at most $kw\text{-}L$, it also holds that either $K(L, S_1)$ or $K(\neg L, S_1)$ holds. Now assume that it holds that $K(L, S_i)$ or $K(\neg L, S_i)$ for some $i \geq 1$. Since $kwL \in S_{i+1}$ the action that was performed in S_i to yield S_{i+1} was either an action that does not change the truth value of L or changes it unconditionally (in other words, it is not an action modified by Step 3 of the compilation). In either case it holds either $K(L, S_{i+1})$ or $K(\neg L, S_{i+1})$. We conclude that $K(L, S_n)$ or $K(\neg L, S_n)$, which means that T achieves $\mathsf{KnowWhether}(L)$. We observe now that if we remove all occurrences of $kw\text{-}act\text{-}pos\text{-}L$ or $kw\text{-}act\text{-}neg\text{-}L$ we obtain a plan that also achieves $\mathsf{KnowWhether}(L)$. ■

Now consider the epistemic goal $\mathsf{Discriminate}(L_1, L_2)$, where L_1 and L_2 are literals. The compilation follows the same intuitions from above. We generate a new transition system $\Sigma' = (F', A', \Omega, I)$, where and F' and A' are computed by performing the following steps:

1. Let F' be $F \cup \{disc\text{-}L_1\text{-}L_2\}$.
2. Add to A' all actions in A plus actions $disc\text{-}act\text{-}1\text{-}L_1\text{-}L_2$ and $disc\text{-}act\text{-}2\text{-}L_1\text{-}L_2$, whose preconditions are $\{L_1, \overline{L_2}\}$ and $\{\overline{L_1}, L_2\}$, respectively. Both actions have a single effect, $disc\text{-}L_1\text{-}L_2$.
3. For each action a in A' that contains $C \to L_1$, $C \to \overline{L_1}$, $C \to L_2$, or $C \to \overline{L_2}$ as a conditional effect, for some C, we add the unconditional effect $\neg disc\text{-}L_1\text{-}L_2$ to a.

This translation is also sound and complete. The proofs are similar to the ones presented above.

Theorem 6 (Completeness) *Let $\Sigma = (F, A, \Omega, I)$ be a transition system, and let Σ' be defined as above. If T is a plan for $(\Sigma, \mathsf{Discriminate}(L_1, L_2)))$ then there exists a plan T' for $(\Sigma', disc\text{-}L_1\text{-}L_2)$ which differs from T only in that it contains actions of the form $disc\text{-}act\text{-}1\text{-}L_1\text{-}L_2$ or $disc\text{-}act\text{-}2\text{-}L_1\text{-}L_2$ at the end of each of T''s branches.*

Theorem 7 (Soundness) *Let $\Sigma = (F, A, \Omega, I)$ be a transition system, and let Σ' be defined as above. If T is a plan for $(\Sigma', \mathsf{Discriminate}(L_1, L_2))$ then, by removing all* *actions of the form* $disc\text{-}act\text{-}1\text{-}L_1\text{-}L_2$ *or* $disc\text{-}act\text{-}2\text{-}L_1\text{-}L_2$ *from* T *we obtain an action tree that is a plan for* $(\Sigma, \mathsf{Discriminate}(L_1, L_2))$.

Extending the Compilation to Formulae We have discussed how to compile goals of the form $\mathsf{KnowWhether}(L)$ and $\mathsf{Discriminate}(L_1, L_2)$, for the case of literals. To extend our compilation to general formulae we can use a pre-compilation step in which, for each formula ϕ involved in the epistemic goal, we generate the ramification constraints ϕ **causes** L_ϕ and $\neg\phi$ **causes** $\neg L_\phi$. Then we apply the compilation we described above. Soundness and completeness now follows from the theorems above together with soundness of the compilation of ramifications into effect axioms.

6 Computing Diagnostic Plans

The purpose of this paper was largely to define the theoretical underpinnings that support the exploitation of state-of-the-art AI planning systems for diagnostic problem solving. As such, our translation can be exploited by any planning system capable of handling conditional effects and sensing actions; in particular it can be used along with conditional planners such as Contingent-FF (Hoffmann and Brafman 2005), POND (Bryce, Kambhampati, and Smith 2006), and the CNF- and DNF-based conditional planners by To, Pontelli, and Son (2011).

Our translation could also be used with the latest contingent online planning systems CLG (Albore, Palacios, and Geffner 2009), CLG+ (Albore and Geffner 2009), and the SDR planner by (Brafman and Shani 2012), even though some of these systems may not return a complete conditional plan. However, for our experiments we wished to generate offline conditional plans and we opted to use Contingent-FF.

We developed 8 diagnostic planning problem scenarios in the following domains: an extension of the light-switch-battery domain given in the example presented in Section 3, an agent moving between rooms fixing light bulbs to complete a circuit, and embedded computer chips. We varied the complexity of the problems and ran them with Contingent-FF using a PC with an Intel Xeon 2.66 GHz processor with 4GB RAM running Linux. The simple problems generated reasonable plans, requiring from under 0.01 seconds to 0.03 seconds to complete. The more complex problems resulted in poor quality plans that favoured conformant rather than contingent solutions. We attribute much of this poor quality to the general effectiveness of the planner and how it's underlying heuristics interact with our approach to handling ramification constraints. We also experimented with encoding ramifications as additional actions rather than compiling them into effects. In future work, we will explore different representations for our ramification constraints across different planners.

An interesting result came from the problems created for the embedded computer chips domain, which consists of a collection of computer chips which themselves contain chips, and so forth, where all chips must be working normally for the output to be displayed. When allowing the planner to sense the status of and replace any chip, the resulting plan was always to replace the top level chips. This supports the notion of taking a purposeful view of diagnosis – it can be much faster to simply repair the issue than to determine the unique diagnosis. More generally, selecting the best course of purposeful actions can be informed by many factor including the cost of actions (sensing and world altering), time-criticality of a response, and issues of specificity or generality with respect to how a grouping of related faults should be addressed (e.g., fix them individually vs. doing a more general fix). The tradeoff between reasoning and acting, and the value of information has been addressed by a variety of researchers, notably Horvitz and Seiver (1997).

7 Related Work and Concluding Remarks

In addition to the literature on model-based diagnosis cited in Section 3, there is a body of previous work that relates diagnosis to theories of action in some guise. The relationship between actions and diagnoses was observed sometime ago by (e.g., Cordier and Thiébaux 1994; McIlraith 1994), while Sampath et al. (1995) were the first to present comprehensive results diagnosing discrete event systems via finite state automata. Thielscher (1997), McIlraith (1998), and subsequently Iwan (2001) and Baral, McIlraith, and Son (2000) cast the diagnosis problems in terms of an AI theory of action and change. More recently Grastien et al. (2007), Sohrabi, Baier, and McIlraith (2010), Lamperti and Zanella (2011) and Yu, Wen, and Liu (2013) have all addressed aspects of dynamical diagnosis.

Specifically in the area of diagnostic and repair planning, the most notable related work is that of Baral, McIlraith, and Son (2000) who introduced the notion of diagnostic and repair plans as conditional plans, and who have a treatment of causality and sensing. That work shares a number of intuitions with the work presented here, but without the focus on planning for epistemic goals, and the correspondence to conformant and contingent planning. Also of note is the work of Kuhn et al. (2008) who introduce the notion of *pervasive diagnosis* which produces diagnostic production plans that achieve production goals while coincidentally uncovering additional information about system health. The notion of combining diagnosis with repair has been addressed in varying fashion by Sun and Weld (1993) and by Friedrich, Gottlob, and Nejdl (1992) among others. Thiébaux et al. (1996), in this and later work, discuss the challenges of diagnosis and repair in the context of a power supply restoration problem, identifying the task as problem of planning with uncertainty. These works share intuitions with the approach advocated here, but do not reflect the advances in our collective understanding of the representational and computational issues associated with planning and sensing. Finally, outside the area of diagnostic problem solving there has been a variety of work looking at planning with some form of sensing. Of particular note is the work of Brenner and Nebel (2009) on MAPL, a continual planning system that interleaves planning with acting and sensing. This paradigm of planning and sensing is also one that is very amenable to diagnostic problem solving.

In this paper we have argued for and explored a purposeful view of diagnostic problem solving, examining the problem through the lens of AI automated planning. We have characterized diagnostic planning with both ontic and epistemic goals, and established properties of these diagnostic planning tasks, including both complexity results and an understanding of their relationship to classical, conformant, and conditional planning systems. Of particular note was the characterization of diagnostic epistemic goals such as Discriminate and KnowWhether and their translation into planning problems with ontic goals. The correspondence with existing planning paradigms enables diagnostic planning to leverage ongoing advances in the development of non-classical planners. We discuss the exploitation of such planners, outlining our experience addressing diagnostic problem solving with Contingent-FF. Results to date are guardedly encouraging but expose the need for further investigation of the nuances of these planners as a complement to the results of this paper.

Beyond diagnostic problem solving, the work presented is relevant to a diversity of problems that involve generating hypotheses to conjecture system state, and sensing and acting in the world to discriminate those hypotheses or to purposefully effect change in response to observed behaviour. Some such problems include active vision applications, activity recognition, and goal recognition.

Acknowledgements: We gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Albore, A., and Geffner, H. 2009. Acting in partially observable environments when achievement of the goal cannot be guaranteed. In *Proc. of ICAPS Workshop on Planning and Plan Execution for Real-World Systems*.

Albore, A.; Palacios, H.; and Geffner, H. 2009. A translation-based approach to contingent planning. In *Proc. of the 21st Int'l Joint Conference on Artificial Intelligence (IJCAI)*, 1623–1628.

Andersen, M. B.; Bolander, T.; and Jensen, M. H. 2012. Conditional epistemic planning. In *Proc. of the 13th European Conf. Logics in Artificial Intelligence (JELIA)*, volume 7519 of *Lecture notes in Computer Science*. Springer. 94–106.

Aucher, G., and Bolander, T. 2013. Undecidability in epistemic planning. In *Proc. of the 23rd Int'l Joint Conference on Artificial Intelligence (IJCAI)*.

Baral, C.; McIlraith, S.; and Son, T. 2000. Formulating diagnostic problem solving using an action language with narratives and sensing. In *Proc. of the 7th Int'l Conference on Knowledge Representation and Reasoning (KR)*, 311–322.

Brafman, R. I., and Shani, G. 2012. Replanning in domains with partial information and sensing actions. *Journal of Artificial Intelligence Research* 45:565–600.

Brenner, M., and Nebel, B. 2009. Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems* 19(3):297–331.

Bryce, D.; Kambhampati, S.; and Smith, D. E. 2006. Planning graph heuristics for belief space search. *J. Artif. Intell. Res. (JAIR)* 26:35–99.

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *Artificial Intelligence* 69(1-2):165–204.

Console, L., and Torasso, P. 2006. Automated diagnosis. *Intelligenza Artificiale* 3(1-2):42–48.

Cordier, M.-O., and Thiébaux, S. 1994. Event-based diagnosis of evolutive systems. In *Proc. of the 5th Int'l Workshop on Principles of Diagnosis (DX)*, 64–69.

de Kleer, J., and Williams, B. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32:97–130.

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

Demolombe, R., and Pozos Parra, M. d. P. 2000. A simple and tractable extension of situation calculus to epistemic logic. In *Proc. of the 12th International Symposium on Foundations of Intelligent Systems*, volume 1932 of *Lecture notes in Computer Science*. Springer. 515–524.

Friedrich, G.; Gottlob, G.; and Nejdl, W. 1992. Formalizing the repair process. In *Proc. of the 10th European Conference on Artificial Intelligence (ECAI)*, 709–713.

Genesereth, M. 1984. The use of design descriptions in automated diagnosis. *Artificial Intelligence* 24:411–436.

Grastien, A.; Anbulagan; Rintanen, J.; and Kelareva, E. 2007. Diagnosis of discrete-event systems using satisfiability algorithms. In *Proc. of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, 305–310.

Haslum, P., and Jonsson, P. 1999. Some results on the complexity of planning with incomplete information. In *Proc. of the 5th European Conference on Planning (ECP)*, 308–318.

Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

Herzig, A.; Lang, J.; and Marquis, P. 2003. Action representation and partially observable planning using epistemic logic. In *Proc. of the 18th Int'l Joint Conference on Artificial Intelligence (IJCAI)*, 1067–1072.

Hoffmann, J., and Brafman, R. I. 2005. Contingent planning via heuristic forward search witn implicit belief states. In *Proc. of the 15th Int'l Conference on Automated Planning and Scheduling (ICAPS)*, 71–80.

Horvitz, E., and Seiver, A. 1997. Time-critical reasoning: Representations and application. In *Proceedings of the Thirteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, 250–257. San Francisco, CA: Morgan Kaufmann.

Iwan, G. 2001. History-based diagnosis templates in the framework of the situation calculus. In *Proc. of the Joint German/Austrian Conference on Artificial Intelligence (KR/ÖGAI)*, 244–259.

Kuhn, L. D.; Price, B.; de Kleer, J.; Do, M. B.; and Zhou, R. 2008. Pervasive diagnosis: The integration of diagnostic goals into production plans. In *Proc. of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, 1306–1312.

Lamperti, G., and Zanella, M. 2011. Context-sensitive diagnosis of discrete-event systems. In *Proc. of the 22nd Int'l*

Joint Conference on Artificial Intelligence (IJCAI), 969–975.

Lin, F. 1995. Embracing causality in specifying the indirect effects of actions. In *Proc. of the 14th Int'l Joint Conference on Artificial Intelligence (IJCAI)*, 1985–1991.

McCain, N., and Turner, H. 1995. A causal theory of ramifications and qualifications. In *Proc. of the 14th Int'l Joint Conference on Artificial Intelligence (IJCAI)*, 1978–1984.

McIlraith, S. A., and Scherl, R. B. 2000. What sensing tells us: Towards a formal theory of testing for dynamical systems. In *Proc. of the 17th National Conference on Artificial Intelligence (AAAI)*, 483–490.

McIlraith, S. 1994. Towards a theory of diagnosis, testing and repair. In *Proc. of the 5th Int'l Workshop on Principles of Diagnosis (DX)*, 185–192.

McIlraith, S. 1997. Representing actions and state constraints in model-based diagnosis. In *Proc. of the 14th National Conference on Artificial Intelligence (AAAI)*, 43–49.

McIlraith, S. 1998. Explanatory diagnosis: Conjecturing actions to explain observations. In *Proc. of the 6th Int'l Conference of Knowledge Representation and Reasoning (KR)*, 167–179.

McIlraith, S. 2000. Integrating actions and state constraints: A closed-form solution to the ramification problem (sometimes). *Artificial Intelligence* 116(1-2):87–121.

Moore, R. C. 1985. A Formal Theory of Knowledge and Action. In Hobbs, J. B., and Moore, R. C., eds., *Formal Theories of the Commonsense World*. Norwood, New Jersey: Ablex Publishing Corp. chapter 9, 319–358.

Palacios, H., and Geffner, H. 2006. Compiling uncertainty away: Solving conformant planning problems using a classical planner (sometimes). In *Proc. of the 21st National Conference on Artificial Intelligence (AAAI)*.

Palacios, H., and Geffner, H. 2009. Compiling uncertainty away in conformant planning problems with bounded width. *Journal of Artificial Intelligence Research* 35:623–675.

Pednault, E. P. D. 1989. ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proc. of the 1st Int'l Conference of Knowledge Representation and Reasoning (KR)*, 324–332.

Petrick, R. P. A., and Bacchus, F. 2002. A knowledge-based approach to planning with incomplete information and sensing. In *Proc. of the 6th Int'l Conference on Artificial Intelligence Planning and Scheduling (AIPS)*, 212–222.

Pinto, J. 1999. Compiling ramification constraints into effect axioms. *Computational Intelligence* 15:280–307.

Pistore, M.; Marconi, A.; Bertoli, P.; and Traverso, P. 2005. Automated composition of web services by planning at the knowledge level. In *Proc. of the 19th Int'l Joint Conference on Artificial Intelligence (IJCAI)*, 1252–1259.

Poole, D. 1994. Representing diagnosis knowledge. *Ann. Math. Artif. Intell.* 11(1-4):33–50.

Pryor, L., and Collins, G. 1996. Planning for contingencies: A decision-based approach. *Journal of Artificial Intelligence Research* 4:287–339.

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

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed., *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*. San Diego, CA: Academic Press. 359–380.

Rintanen, J. 2004. Complexity of planning with partial observability. In *Proc. of the 14th Int'l Conference on Automated Planning and Scheduling (ICAPS)*, 345–354.

Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; and Teneketzis, D. 1995. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control* 40(9):1555–1575.

Sandewall, E. 1996. Assessments of ramification methods that use static domain constraints. In *Proc. of the 5st Int'l Conference of Knowledge Representation and Reasoning (KR)*, 99–110.

Scherl, R., and Levesque, H. 2003. Knowledge, action, and the frame problem. *Artificial Intelligence* 144(1–2):1–39.

Shortliffe, E. H., and Buchanan, B. G. 1975. A model of inexact reasoning in medicine. *Mathematical Biosciences* 23(34):351 – 379.

Sohrabi, S.; Baier, J. A.; and McIlraith, S. 2010. Diagnosis as planning revisited. In *Proc. of the 12th Int'l Conference on the Principles of Knowledge Representation and Reasoning (KR)*, 26–36.

Sohrabi, S.; Baier, J. A.; and McIlraith, S. 2011. Preferred explanations: Theory and generation via planning. In *Proc. of the 25th AAAI Conference on Artificial Intelligence (AAAI)*.

Strass, H., and Thielscher, M. 2013. A general first-order solution to the ramification problem with cycles. *Journal of Applied Logic* 11(3):289–308.

Sun, Y., and Weld, D. S. 1993. A framework for model-based repair. In *Proc. of the 11th National Conference on Artificial Intelligence (AAAI)*, 182–187.

Thiébaux, S.; Cordier, M.-O.; Jehl, O.; and Krivine, J.-P. 1996. Supply restoration in power distribution systems: A case study in integrating model-based diagnosis and repair planning. In *Proc. of the 12th Conference in Uncertainty in Artificial Intelligence (UAI)*, 525–532.

Thielscher, M. 1995. Computing ramifications by postprocessing. In *Proc. of the 14th Int'l Joint Conference on Artificial Intelligence (IJCAI)*, 1994–2000.

Thielscher, M. 1997. A theory of dynamic diagnosis. *Electronic Transactions on Artificial Intelligence* 1:73–104.

To, S. T.; Pontelli, E.; and Son, T. C. 2011. On the effectiveness of CNF and DNF representations in contingent planning. In *Proc. of the 22nd Int'l Joint Conference on Artificial Intelligence (IJCAI)*, 2033–2038.

Yu, Q.; Wen, X.; and Liu, Y. 2013. Multi-agent epistemic explanatory diagnosis via reasoning about actions. In *Proc. of the 23rd Int'l Joint Conference on Artificial Intelligence (IJCAI)*.