Behavior Interpretation:
Making sense of people, machines, and the world around us

Sheila A. McIlraith
Dept. of Computer Science, University of Toronto
Vector Institute, Toronto

PAIR2020 at AAAI 2020
New York, NY
February 8, 2020

Computer Science
UNIVERSITY OF TORONTO

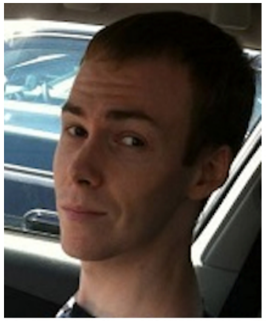
VECTOR INSTITUTE | INSTITUT VECTEUR

Acknowledgements

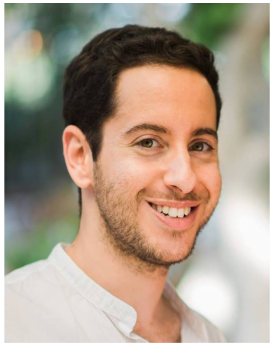
Jorge Baier

Shirin Sohrabi

Christian Fritz



Brent Mombourquette



Maayan Shvo

Motivation

Understanding the behavior of people, machines, and our environment is something we as humans do continuously. Sometimes we do so to satisfy our curiosity, but much of the time it's purposeful -- so that we can decide how to act.

Our ability to make sense of observed behavior is informed by our expectations of the behavior of the actor and by our observations, and the quality of the conclusions we draw relies heavily on the quality of these two elements.

In many cases we have agency to actively sense and manipulate the world in service of this pursuit. Whether it's goal recognition, plan recognition, narrative understanding, video interpretation, or diagnostic problem solving, the patterns of inference are similar.

In this talk I'll cast these somewhat disparate tasks as instances of the general problem of behavior interpretation and discuss the elements that inform its effective realization. The discussion will be informed by past and ongoing work in my research group on diagnostic problem solving and goal and plan recognition.

Plan Recognition

Goal Recognition

Explanation Generation

Automated Diagnosis

Video Analysis

Behavior Interpretation

Activity Recognition

Intent Recognition

Narrative Understanding

Auditing/Monitoring of Business Processes

Environmental Interpretation

about people

about machines/devices

about the world around us

Plan Recognition

Goal Recognition

Explanation Generation

Automated Diagnosis

Video Analysis

Behavior Interpretation

Activity Recognition

Intent Recognition

Narrative Understanding

Auditing/Monitoring of Business Processes

Environmental Interpretation

about people **about machines/devices** **about the world around us**

Plan Recognition

Goal Recognition

Explanation Generation

Automated Diagnosis

Video Analysis

Behavior Interpretation

Activity Recognition

Intent Recognition

Narrative Understanding

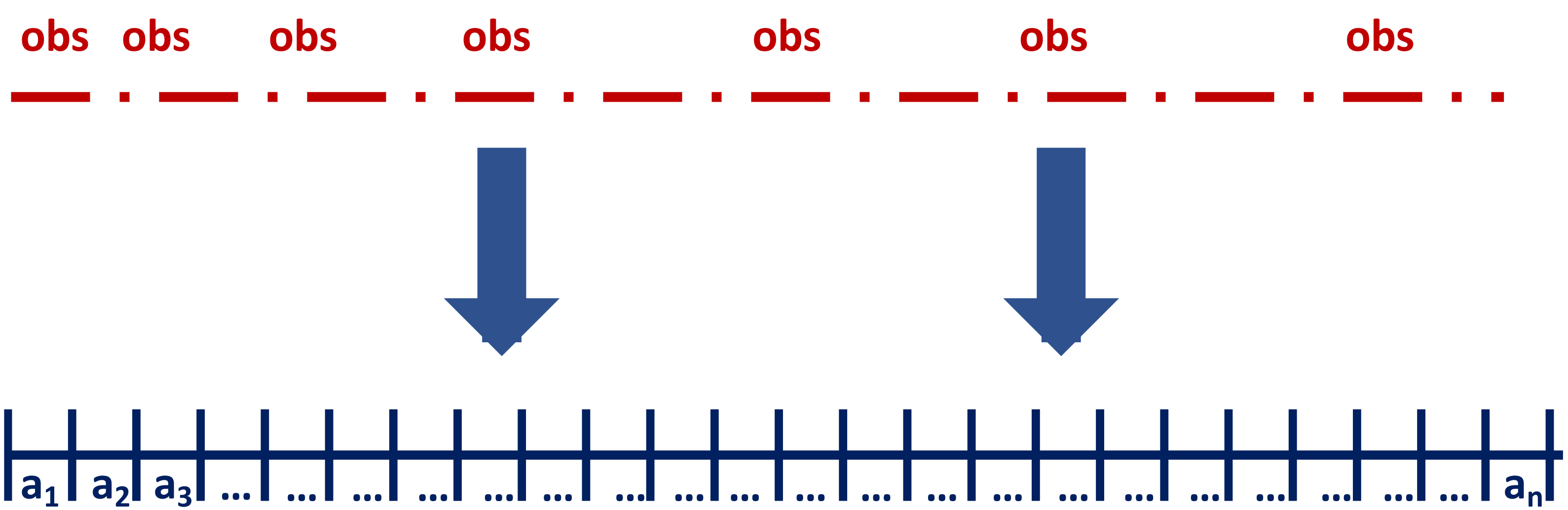
Auditing/Monitoring of Business Processes

Environmental Interpretation

Pattern of Inference

Align

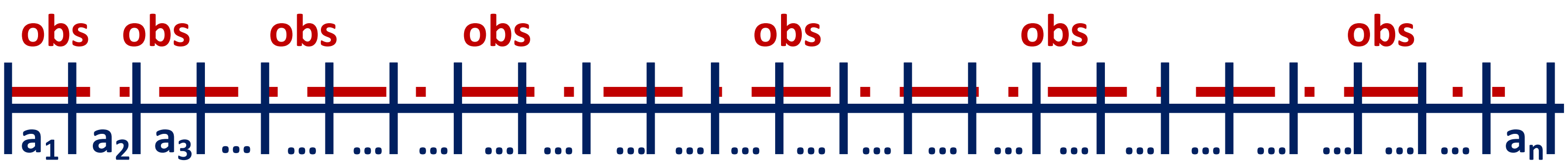
- **observations** realized over time, with
- some **expectation of behavior**



Pattern of Inference

Align

- **observations** realized over time, with
- some **expectation of behavior**



Relationship to Planning

Various tasks related to behavior interpretation can be realized by AI planning:

- Non-classical planning
- Conformant planning
- Conditional planning (i.e., offline contingent planning)
- Contingent planning
- Epistemic Planning

Relationship to Planning

Various tasks related to behavior interpretation can be realized by AI planning:

- Non-classical planning
- Conformant planning
- Conditional planning (i.e., offline contingent planning)
- Contingent planning
- Epistemic Planning

Continuing advances in planning technologies are enabling us to revisit and make progress on tasks that historically we understood how to specify but were unable to realize computationally. We see this in some of the work that follows.

There are a bunch of moving parts

- What **model**(s) (or **data**) do you use to generate expectations?
- What are your **observations**? Are they regular, sufficient? Are they noisy?
- What **constitutes an alignment** (consistency, entailment, …)
- What does the **solution** look like?
- What **purpose** does/should the solution serve?
- What **computational machinery** are you using?
- Is the interpretation done **post hoc or online**?

Intermission

About the intermission …

This was the "magic show" where the diagnosis of a malfunctioning flashlight was used to interactively demonstrate concepts related to automated diagnosis, goal recognition, and active behavior interpretation more generally.

Observations from the flashlight example

- Goal/Plan recognition can be done **post hoc or online**
- **Observations of action** *and* **state** are both relevant & useful
- **Goals** can be **temporally extended**
- **Goals** can be **epistemic**
- The actor (me) **used beliefs about the observers' (changing) models to realize her goal**
- The **observer can have agency** to **sense/reason/act** to expedite or make possible recognition or to assist or impede goal realization
- It's important to model the **actions** in the context of the **environment**
- The recognition task is often **purposeful** – you need not find a unique answer/solution. Often one need only discriminate sufficiently to **decide how to act**.

The Rest of This Talk

I. Diagnosis as Planning
II. Diagnostic Problem Solving (and the role of epistemics)
III. What Sensing Tells Us (and the notion of tests)

I. Diagnosis as Planning

(Some*) previous work on diagnosis

..., (Reiter, 1987),
(de Kleer & Williams, 1989),
(Junker, 1991),
(de Kleer, Mackworth, & Reiter, 1992),
(Torta & Torasso, 2004),
(Cordier & Thiébaux, 1994),
(McIlraith, 1994),
(Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995),
(Thielscher, 1997),
(McIlraith, 1998),
(Baral, McIlraith, & Son, 2000),
(Iwan, 2001),
(Lamperti & Zanella, 2003),
(Iwan & Lakemeyer, 2003),
(Mikaelian, Williams, & Sachenbacher, 2005),
(Pencolé & Cordier, 2005),
(Grastien, Anbulagan, Rintanen, & Kelareva, 2007a),
(Rintanen & Grastien, 2007),...

*** And many more. Apologies to those I've missed.**

Diagnosis as Planning [Sohrabi, Baier, M, KR10, AAAI11; M, AAAI97]

Task: Diagnosis of Discrete Dynamical Systems

Given a description of system behaviour and a set of observations, determine **what happened** to the system in terms of actions that have occurred and that can account for the observed behaviour.

Objective:

Leverage state-of-the-art planning for the generation of diagnoses.

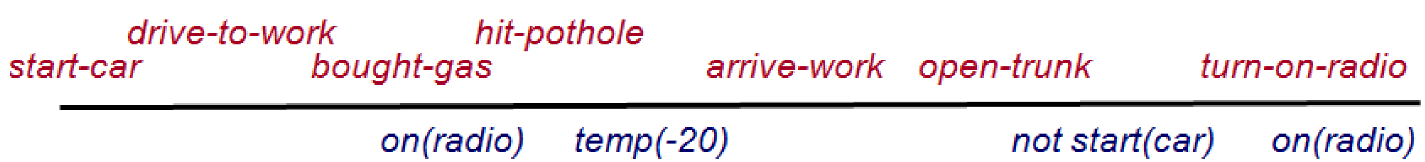
Contributions:

1. Our formal characterization of diagnosis,
2. Correspondence between diagnosis and planning,
3. State-of-the-art planning for the generation of diagnoses.

Example

Observations

I started my car this morning; drove to work; on the way to work I bought $5 worth of gas; I hit a pothole; the radio said it was -20 Celsius; I parked outside. At noon, I picked up my bag from the trunk of the car. At the end of the day, my car would not start. I checked the radio and it was still working.



What's the explanation for my car not starting?

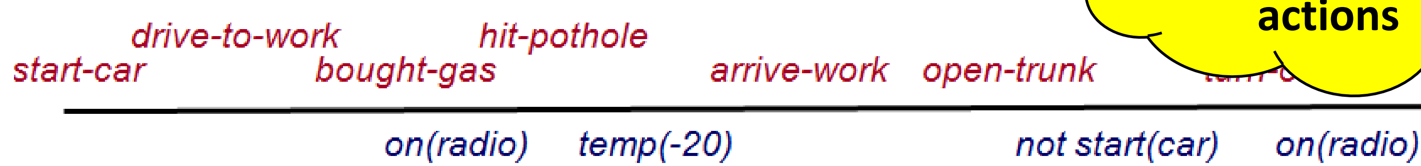
- Battery died
- Punctured gas tank, then ran out of gas
- Starter motor broke

Example

Observations

I started my car this morning; drove to work; on the way to work I bought $5 worth of gas; I hit a pothole; the radio said it was -20 Celsius; I parked outside. At noon, I picked up my bag from th trunk of the car. At the end of the day, my car wou checked the radio and it was still working.

Observations are over state properties and actions



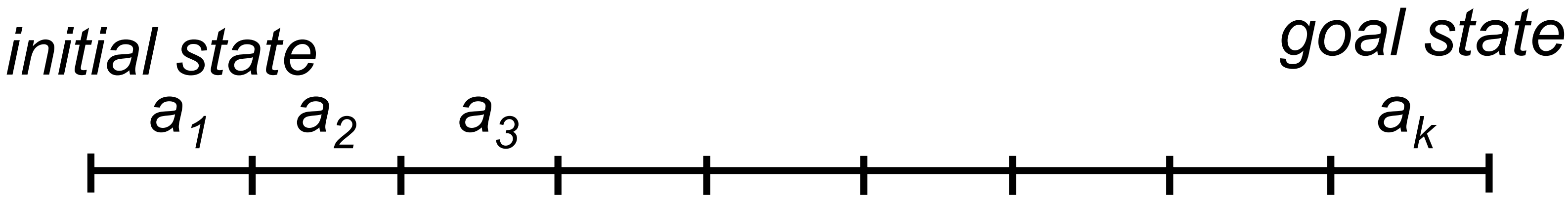
What's the explanation for my car not starting?

- Battery died
- Punctured gas tank, then ran out of gas
- Starter motor broke

Classical Planning 101

Classical Planning

- Initial State
- Goal State
- Transition System



Plan $= \{a_1, a_2, a_3, \ldots, a_k\}$

Diagnosis as Non-Classical Planning

Dynamical Diagnosis

- ~~Initial State~~
- ~~Goal State~~

Observations
(i.e., multiple partial states)

- ~~Transition System~~ System Description

obs_1 a_1 a_2 a_3 obs_2 obs_3 . . . obs_n a_k

Diagnosis = {*Assumptions*, {a_1, a_2, a_3, …, a_k}}

Diagnosis as Non-Classical Planning

Dynamical Diagnosis

- ~~Initial State~~ →
- ~~Goal State~~ →

Observ

(i.e., r

- ~~Transition System~~ → S

Assumptions together with conformant plans guarantee executability and goal realization.

obs_1 a_1 a_2 a_3 *obs_2* *obs_3* . . . *obs_n* a_k

Diagnosis = {Assumptions, {a_1, a_2, a_3, …, a_k}}

Observations in Linear Temporal Logic

Observations play the role of **Temporally Extended Goals** expressed in Linear Temporal Logic (LTL), a compelling logic to express temporal properties of traces.

Syntax

Logic connectives: $\wedge, \vee, \neg$

LTL **basic operators:**

- next: $\bigcirc\varphi$
- weak next: $\bullet\varphi$
- until: $\psi \,\mathsf{U}\, \chi$

Other LTL **operators:**

- eventually: $\Diamond\varphi \stackrel{\text{def}}{=} \text{true}\,\mathsf{U}\,\varphi$
- always: $\Box\varphi \stackrel{\text{def}}{=} \neg\Diamond\neg\varphi$
- release: $\psi \,\mathsf{R}\, \chi \stackrel{\text{def}}{=} \neg(\neg\psi \,\mathsf{U}\, \neg\chi)$

Properties

- Fluents augmented with “occ(a)” to express the occurrence of actions.
- LTL Interpreted over **finite** or **infintite** traces.
- Can be transformed into **automata.**

Explanatory Diagnosis

Use LTL to represent observation

Definition (Explanation)

Given a system $\Sigma = (F, A, I)$, and an observation φ, expressed in LTL an explanation is a tuple (H, α), where

1. H is a set of clauses over F st. $I \cup H$ is satisfiable, $I \not\models H$,
2. $\alpha = a_0 a_1 \dots a_n$, a sequence of actions in A st. α satisfies φ in the system $\Sigma_A = (F, A, I \cup H)$.

Definition (Optimal Explanation)

Given a system Σ, E is an optimal explanation for observation φ iff

1. E is an explanation for φ, and
2. there does not exist another explanation E' for φ st $E' \prec E$.

Relationship to Planning

Proposition

Given a dynamical system Σ and an observation formula φ, then

$$(H, \alpha = a_0 a_1 \ldots a_n) \text{ is an explanation}$$

iff

α *is a plan for* **conformant** *planning problem*
$P = ((F, A, I \cup H), \varphi)$
where $I \cup H$ *is satisfiable and* φ *is a temporally extended goal.*

Theorem

Given a dynamical system Σ and a temporally extended formula φ, explanation existence is PSPACE-complete.

Theorem

It is possible to find explanations using classical planning.

Relationship to Planning

Proposition

Given a dynamical system Σ and an observation formula φ, then

$$(H, \alpha = a_0 a_1 \ldots a_n) \text{ is an explanation}$$

<u>*iff*</u>

α is a plan for **conformant** *planning problem*
$P = ((F, A, I \cup H), \varphi)$
where $I \cup H$ is satisfiable and φ is a temporally extended goal.

Incomplete information about the initial state, so not classical planning.

Σ and a temporally extended formula φ,
SPACE-complete.

It is possible to find explanations using classical planning.

Diagnosis as Planning

See the paper for discussion of computation, experiments, and the use of Past LTL preferences.

The Rest of This Talk

I. Diagnosis as Planning
II. Diagnostic Problem Solving (and the role of epistemics)
III. What Sensing Tells Us (and the notion of tests)

II. Diagnostic Problem Solving (and the Role of Epistemics)

Looking at diagnosis purposefully and the myriad of tasks associated with diagnosis, testing and repair

Be PURPOSEFUL!
What problem(s) do we need to solve?
What's an appropriate "solution"?

Diagnostic Problem Solving: A planning perspective

[Baier, Mombourquette, M, KR14]

Our Contributions

- Discuss reasoning tasks associated with diagnostic problem solving
  - Identify the need for *epistemic goals*
- Map those tasks to offline automated planning
  - Show that epistemic goals can be *compiled away*
- Characterize the complexity of those tasks
- Investigate how state-of-the-art planners scale at these tasks

Diagnosis (following [Reiter, AIJ87])

Static System

A static system is a tuple $(SD, COMPS, OBS)$

- SD: system description
- $COMPS$: set of components
- OBS: observations

 In the flashlight example:

- SD: $on \wedge \neg AB(battery) \wedge \neg AB(switch) \supset light$
- $COMPS$: $\{battery, switch\}$
- OBS: $\{on, \neg light\}$

Minimal Diagnosis

Diagnosis

Given a diagnostic system Σ_{SD}, $\Delta \subseteq COMPS$ is a diagnosis iff

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

is satisfiable.

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.

Dynamism makes problem interesting

In a **dynamic domain**:

- The agent can manipulate the system via actions
- *Repair* actions could be available
- *Sensing* actions could be available
- The best course of action may or may not involve sensing/repairing

Dynamical System Model

- Dynamic systems as transition systems as a tuple $\Sigma = (F, A, \Omega, I)$
- F: set of fluents
- A: set of actions that admit conditional effects of the form $C \rightarrow L$, for C a conjunction of fluent literals and L a fluent literal.
- Ω: set of sensing actions
- I is a boolean formula describing an initial belief state.

- **Propositional** model

Diagnostic Problem Solving

Diagnostic Problem Solving includes diagnosis, testing and repair.

A unique diagnosis may not be the desired outcome.

We may wish to

- Eradicate egregious behavior
- Repair the system
- Confirm/refute specific diagnoses/hypotheses
- Confirm/refute certain properties of the system
- …

Diagnostic System

Given SD, $COMPS$, OBS, and $\Sigma = (F, A, \Omega, I)$, a diagnostic system Σ_{SD} is a tuple (F', A', Ω, I') where:

- $F' = F \cup Vars(SD) \cup \{AB(c)\}$ for all $c \in COMPS$
- $I' = I \cup SD \cup OBS$.
- $A' = A$ but augmented, following (Pinto 99), to address the ramification problem

It's important to model the "environment" (here the device) and how actions indirectly affect many aspects of the environment.

System Description as Action Theory

Input System:

- SD: $on \wedge \neg AB(battery) \wedge \neg AB(switch) \supset light$
- $COMPS$: $\{battery, switch\}$
- OBS: $\{on, \neg light\}$

Initial Action Theory:

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

Augmented Planning Action Theory:

a	$prec(a)$	(some) Additional Effects [via Pinto 99]
turn-on	$\neg on$	$\neg AB(battery) \wedge \neg AB(switch) \rightarrow light$
change-battery	$true$	$on \wedge \neg AB(switch) \rightarrow light$
fix-switch	$true$	$on \wedge \neg AB(battery) \rightarrow light$
sense-light	$true$	

System Description as Action Theory

Input System:

- SD: $on \wedge \neg AB(battery) \wedge \neg AB(switch) \supset light$
- $COMPS$: $\{battery, switch\}$
- OBS: $\{on, \neg light\}$

Initial Action Theory:

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

Indirect Effects of Actions are Important!

Augmented Planning Action Theory:

a	$prec(a)$	(some) Additional Effects [via Pinto 99]
turn-on	$\neg on$	$\neg AB(battery) \wedge \neg AB(switch) \rightarrow light$
change-battery	$true$	$on \wedge \neg AB(switch) \rightarrow light$
fix-switch	$true$	$on \wedge \neg AB(battery) \rightarrow light$
sense-light	$true$	

Planning Task Characterization

- Plans are defined as **action trees**:

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

- ϵ; 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.

- Given action tree T, we say that

$$(T, S) \vdash (T', S')$$

iff
executing one step of T in S leads you to S' with T' remaining,
where S is a *set* of states

Planning Task Characteri

Our diagnostic problem solving tasks requiring sensing are conditional planning task (i.e., offline contingent planning)

- Plans are defined as **action trees**:

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

- ϵ; 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.

- Given action tree T, we say that

$$(T, S) \vdash (T', S')$$

iff

executing one step of T in S leads you to S' with T' remaining, where S is a *set* of states

Diagnostic Plan

Given a diagnostic planning task $(\Sigma_{SD}, Init, \Phi, G)$ where:

- $\Sigma_{SD} = (F, A, \Omega, I)$, the diagnostic system
- $Init$, additional initial state information
- Φ, logical formula representing state constraints
- G, diagnostic planning goal literals

Action tree T is a diagnostic plan for $(\Sigma_{SD}, Init, \Phi, G)$ under the constraint Φ iff

- For every S such that $(T, S_0) \vdash^* (\epsilon, S)$, $s \models G$, for every $s \in S$.
- For every S such that $(T, S_0) \vdash^* (T', S)$, $s \models \Phi$, for every $s \in S$.

Diagnostic Problem Solving Tasks

- **Eradicate Egregious Behaviour**

  "I want the light to be on"

- **Given a diagnosis Δ, do a repair**
- **Know whether a certain diagnosis Δ**

  "I want a plan to know whether John has an allergy to X"

- **Discriminate between two diagnosis Δ_1, Δ_2**

  "Generate a plan to determine if John has a bacterial or fungal infection"

- **Purposeful View**.

  Do not necessarily give me a diagnosis. I care about acting.

Diagnostic Problem Solving Tasks

- **Eradicate Egregious Behaviour**

$$G = \neg OBS$$

- **Given a diagnosis Δ, do a repair**

$$Init = \{AB(\Delta)\} \cup \{\neg AB(COMPS \setminus \Delta)\}$$

$$G = \neg AB(\Delta)$$

- **Know whether a certain diagnosis Δ**
  $G = \textbf{KnowWhether}(AB(\Delta))$

- **Discriminate between two diagnosis Δ_1, Δ_2**
  $G = \textbf{Discriminate}(AB(\Delta_1), AB(\Delta_2))$

Epistemic Diagnostic Planning

- Belief level planning: states capture agent beliefs
- $K(\phi, S)$ iff $s \models \phi$ for each $s \in S$

Epistemic Plans

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

KnowWhether(L) compiled to Ontic Goal

Compile $\Sigma_{SD} = (F, A, \Omega, I)$ into $\Sigma'_{SD} = (F', A', \Omega, I)$ as

- $F' = F \cup \{kw\text{-}L\}$
- $A' = A \cup \{kw\text{-}act\text{-}pos\text{-}L, kw\text{-}act\text{-}neg\text{-}L\}$ such that
  $eff(kw\text{-}act\text{-}pos\text{-}L) = eff(kw\text{-}act\text{-}neg\text{-}L) = kw\text{-}L$
  $prec(kw\text{-}act\text{-}pos\text{-}L) = L$ and $prec(kw\text{-}act\text{-}neg\text{-}L) = \neg L$
- $\forall a \in A$, if $C \rightarrow L \in a$ or $C \rightarrow \neg L \in a$, add $\neg kw\text{-}L$ as an effect

Theorem (Completeness) [paraphrased]

If T is a plan for $(\Sigma_{SD}, \mathbf{KnowWhether}(L))$ then there exists T' a plan for $(\Sigma_{SD}, kw\text{-}L)$.

Theorem (Soundness) [paraphrased]

If T is a plan for $(\Sigma_{SD}, kw\text{-}L)$ then T', T with *kw-act-pos-L* and *kw-act-neg-L* actions removed, is a plan for $(\Sigma_{SD}, \mathbf{KnowWhether}(L))$.

Discriminate(L1,L2) compiled to Ontic Goal

Compile $\Sigma_{SD} = (F, A, \Omega, I)$ into $\Sigma'_{SD} = (F', A', \Omega, I)$ as

- $F' = F \cup \{disc\text{-}L_1\text{-}L_2\}$
- $A' = A \cup \{disc\text{-}act\text{-}1\text{-}L_1\text{-}L_2, disc\text{-}act\text{-}2\text{-}L_1\text{-}L_2\}$ such that $eff(disc\text{-}act\text{-}1\text{-}L_1\text{-}L_2) = eff(disc\text{-}act\text{-}2\text{-}L_1\text{-}L_2) = disc\text{-}L_1\text{-}L_2$ $prec(disc\text{-}act\text{-}1\text{-}L_1\text{-}L_2) = \{L_1, \neg L2\}$ and $prec(disc\text{-}act\text{-}2\text{-}L_1\text{-}L_2) = \{\neg L_1, L2\}$
- $\forall a \in A$, if $C \rightarrow L_1$, $C \rightarrow \neg L_1$, $C \rightarrow L_2$, or $C \rightarrow \neg L_2$ in a, add $\neg disc\text{-}L_1\text{-}L_2$ as an effect

- Theorems of **Soundness and Completeness** similar to **KnowWhether**(L)

Complexity Results

1. Diagnostic planning with complete information and without sensing is PSPACE-complete. (Follows from Bylander 1994)
2. Diagnostic planning without sensing is EXPSPACE-complete. (Follows from Haslum and Jonsson 1999)
3. Diagnostic planning with sensing is 2-EXPTIME-complete. (Follows from Rintanen 2004)

Diagnostic Problem Solving

We did some interesting experiments with two offline contingent planners (Contingent-FF [Hoffman, Brafman] and later CNF_{CT} and DNF_{CT} [To, Pontelli, Son]. We subsequently did experiments with PO-PRP [Muise, Belle, M] which displayed far superior results.

The Rest of This Talk

I. Diagnosis as Planning
II. Diagnostic Problem Solving (and the role of epistemics)
III. What Sensing Tells Us (and the notion of tests)

III. What Sensing Tells Us (and the notion of tests)

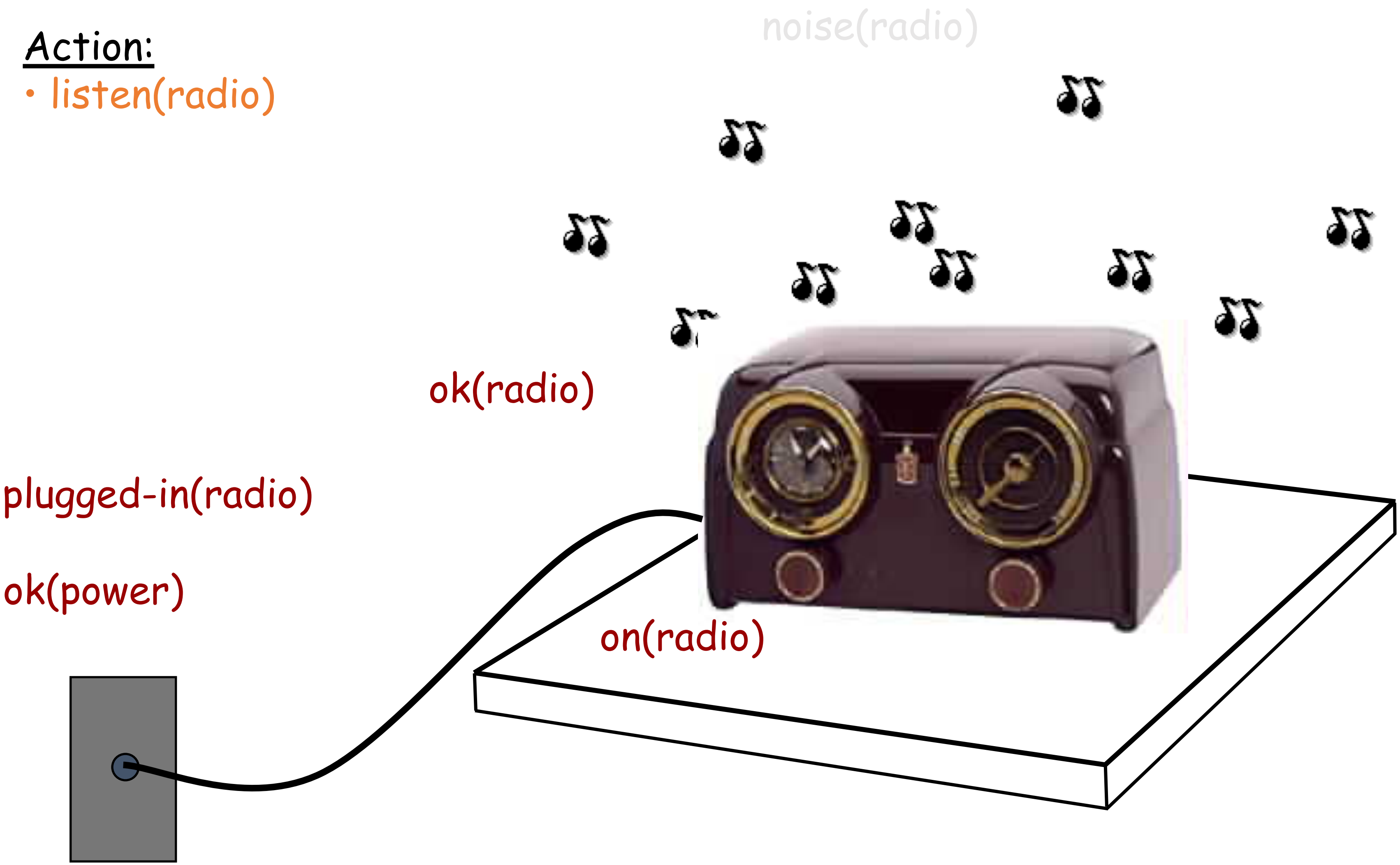
What Sensing Tells Us [M, Scherl, AAAI00, M, Reiter 1992]

Example

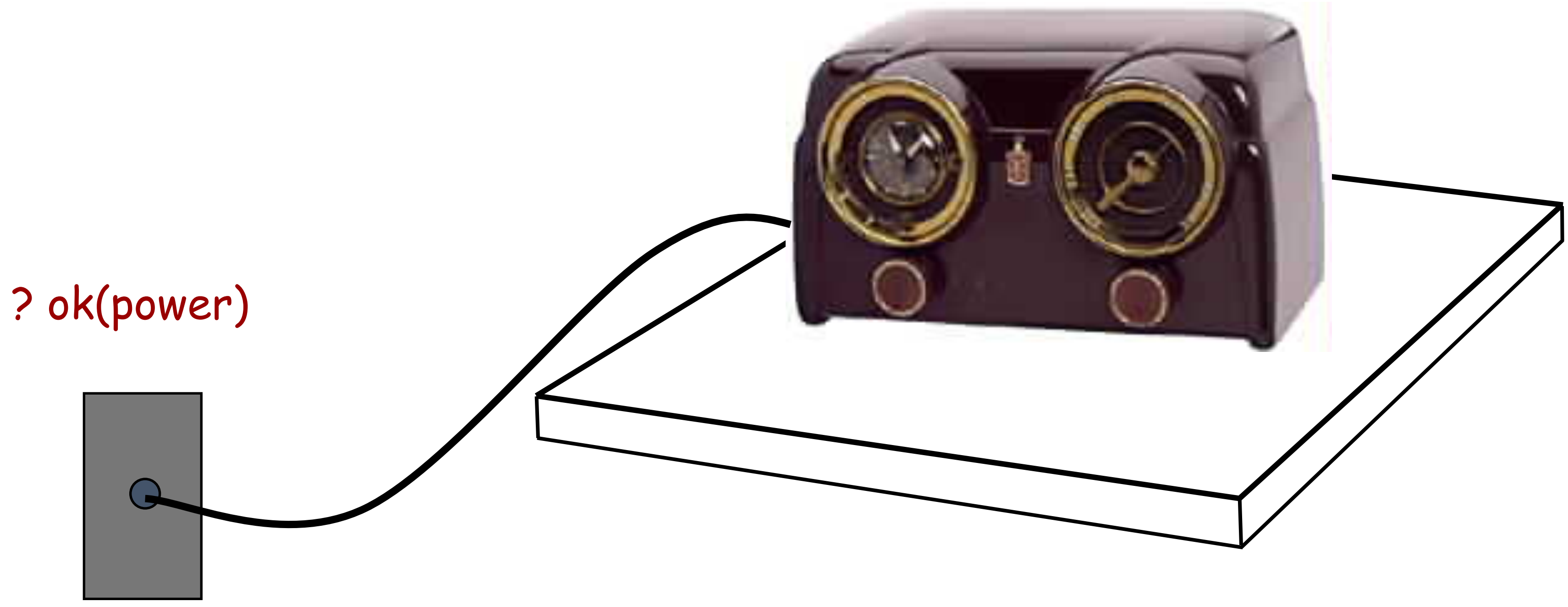
Indirect Effects of "listen(radio)"

Action:

- listen(radio)



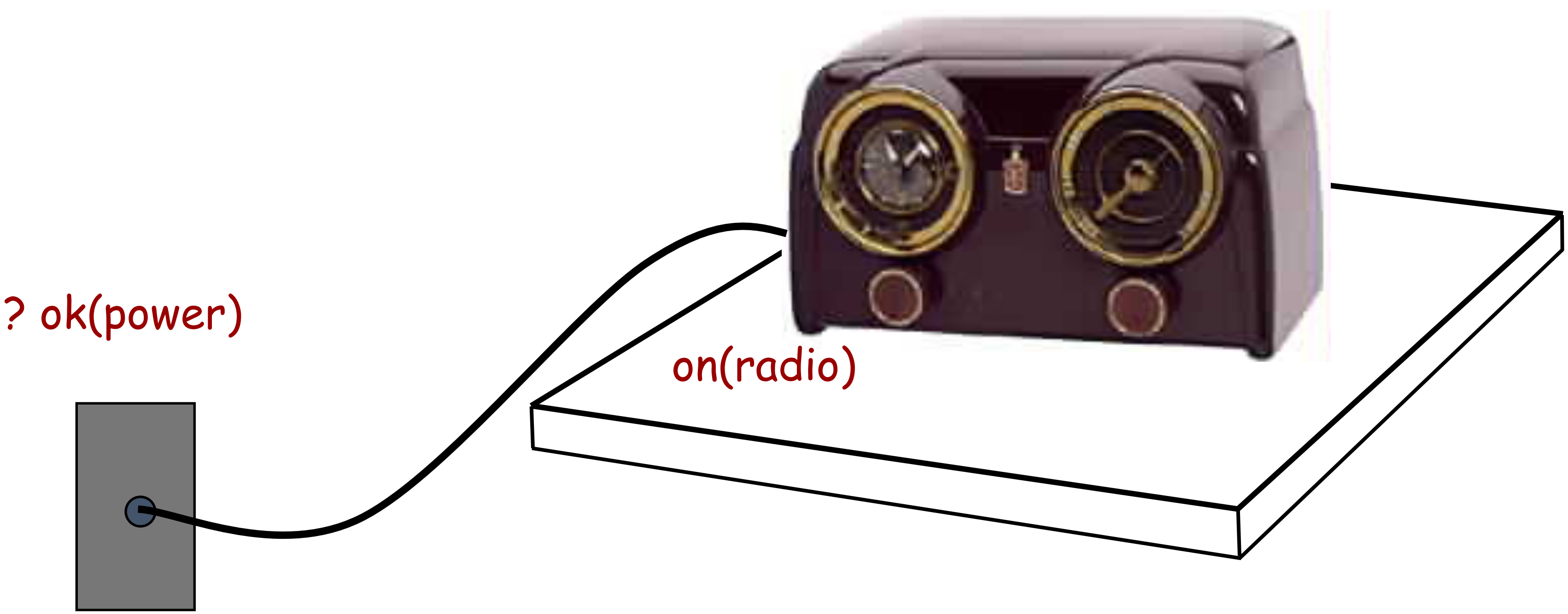
Determine whether "ok(power)"



Determine whether "ok(power)"

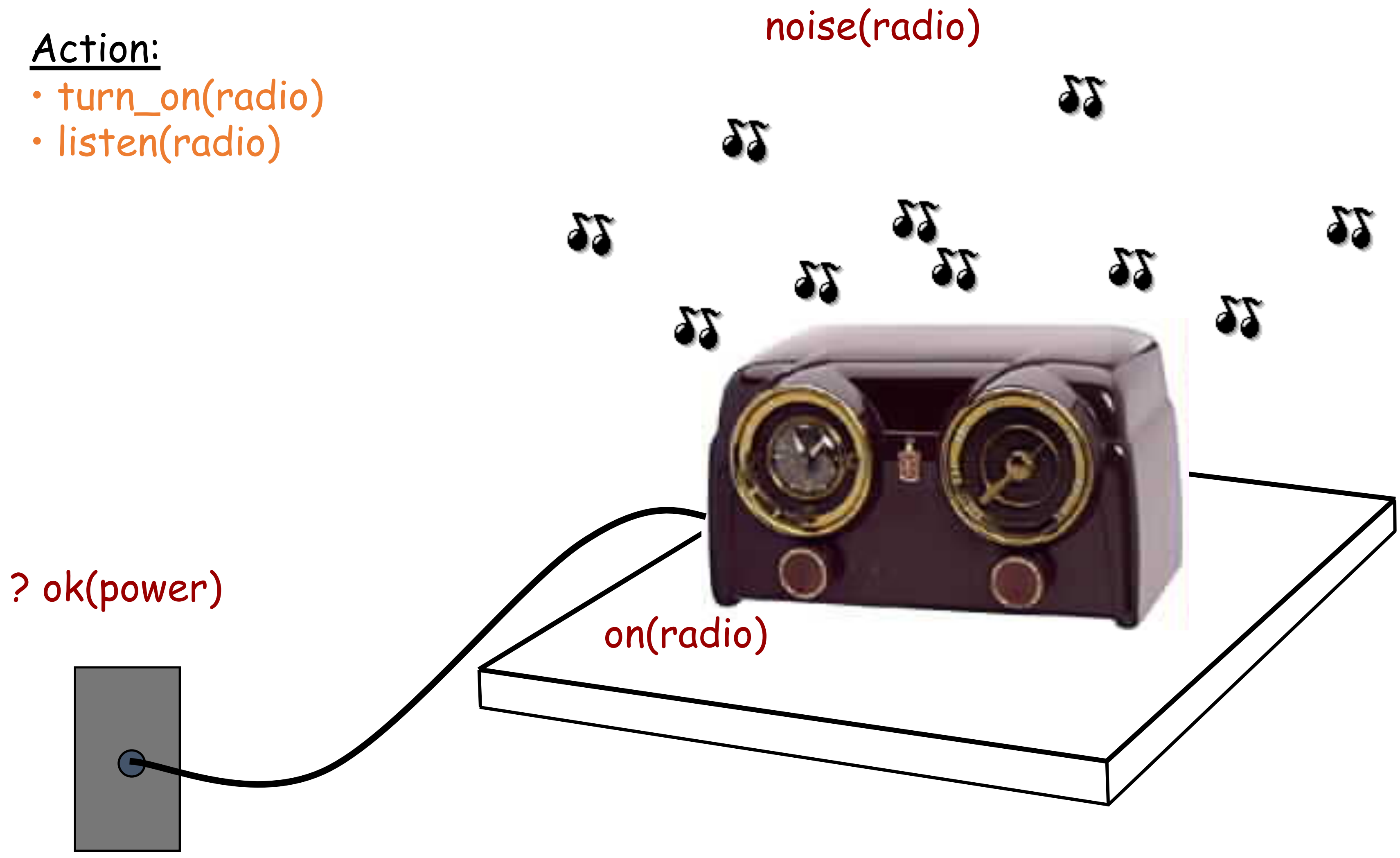
Action:

- turn_on(radio)



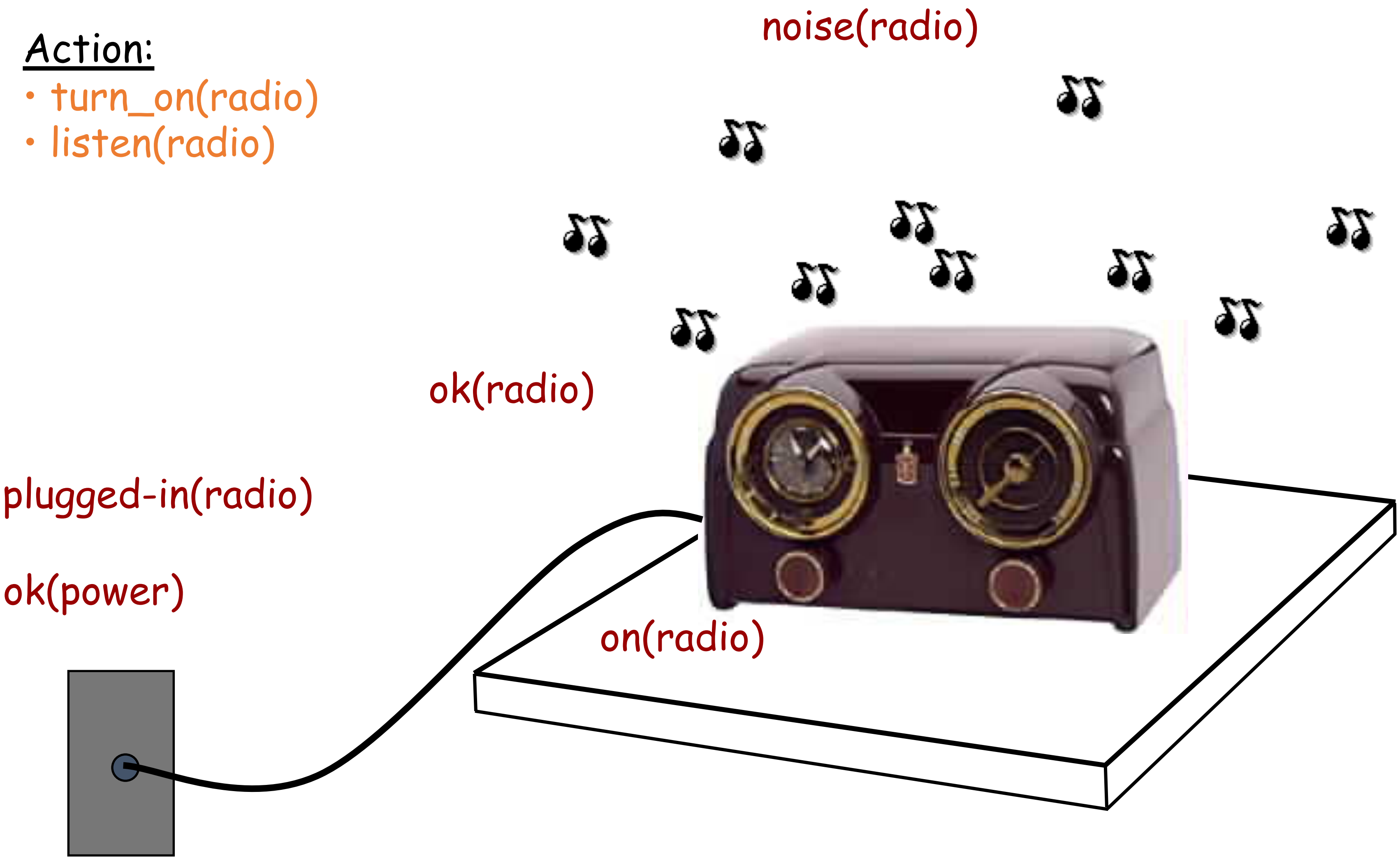
Determine whether "ok(power)"

Action:
- turn_on(radio)
- listen(radio)



Determine whether "ok(power)"

Action:
- turn_on(radio)
- listen(radio)



Determine whether “ok(power)”

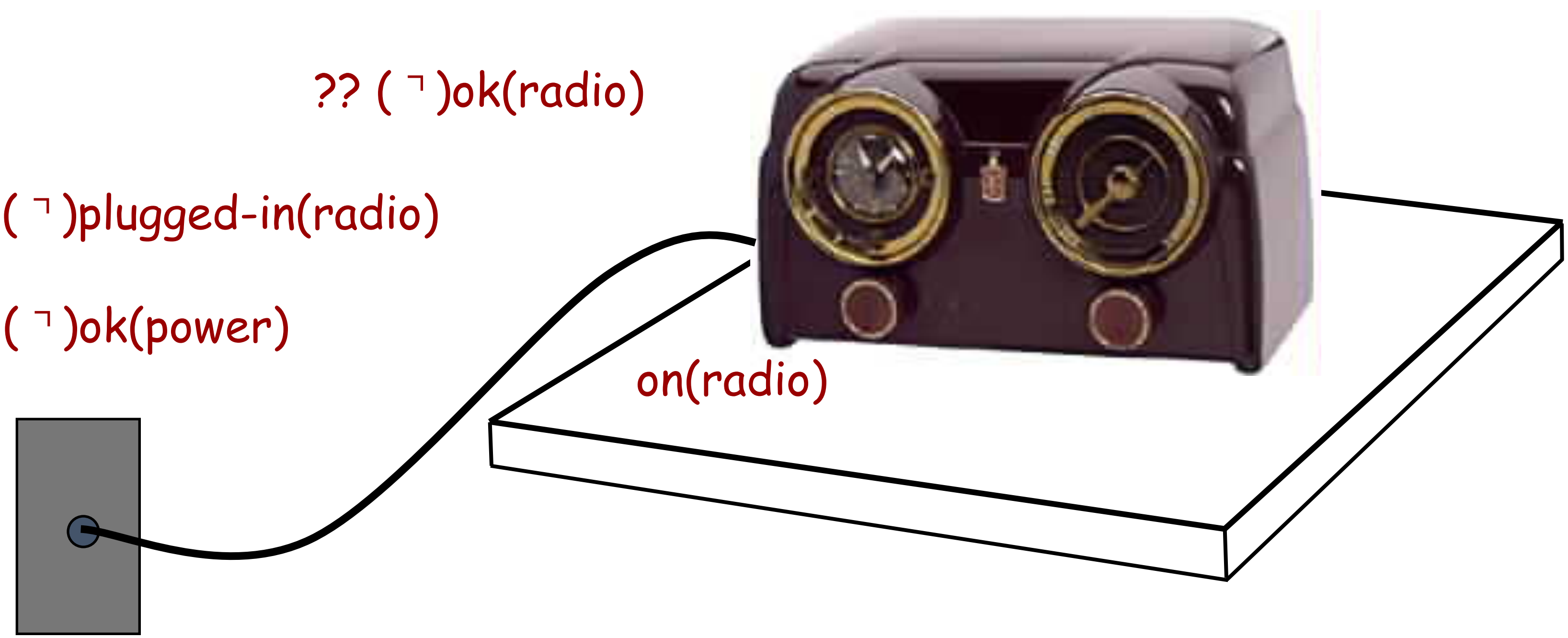
<u>Action:</u>
- turn_on(radio)
- listen(radio)

¬ noise(radio)

… … … … … … …

… … silence … …

… … …



Problem and Approach

Problem:

Given an axiomatization of a deterministic, partially observable dynamical system with

- sensing actions
- state constraints

(relationships between properties/objects in the world).

and a set of unobservable hypotheses

How do we select actions to reduce the hypothesis space?

Approach:

Provide a theory of testing for dynamical systems in the situation calculus.

Contributions

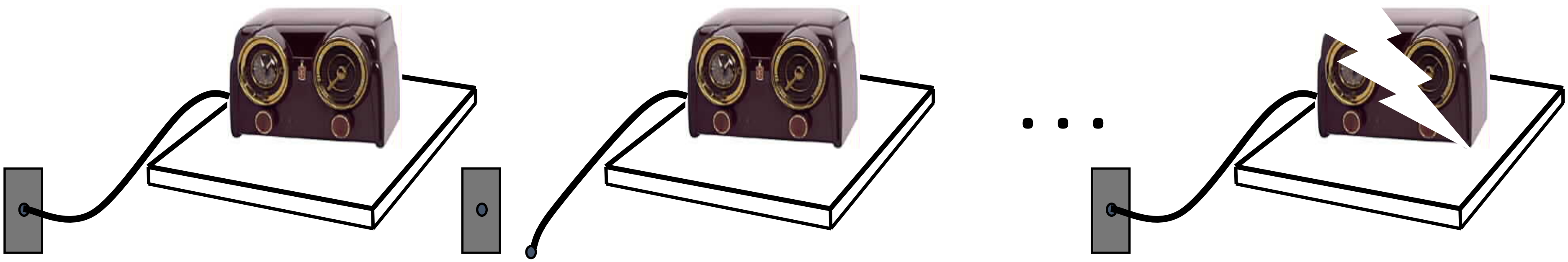
- “Solution” to the ramification problem for sensing actions
- Characterization of tests, and the effect of test outcomes
- Effect of test outcomes on different hypothesis spaces
- Complex tests as Golog procedures
- Verification and generation of complex tests

Ramification Problem for Sensing Actions

Theorem (informally stated):
Our representation addresses the frame and ramification problems for world-altering and sensing actions.

Using this representation the agent knows the indirect effects of both its world-altering and sensing actions.

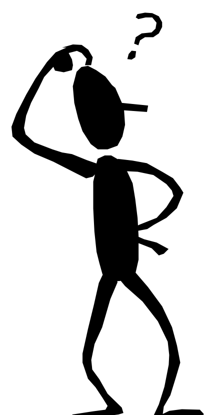
Adding Knowledge and Sensing Actions



¬ ok(power,s)
plugged-in(radio,s)
ok(radio,s)
on(radio,s)
…

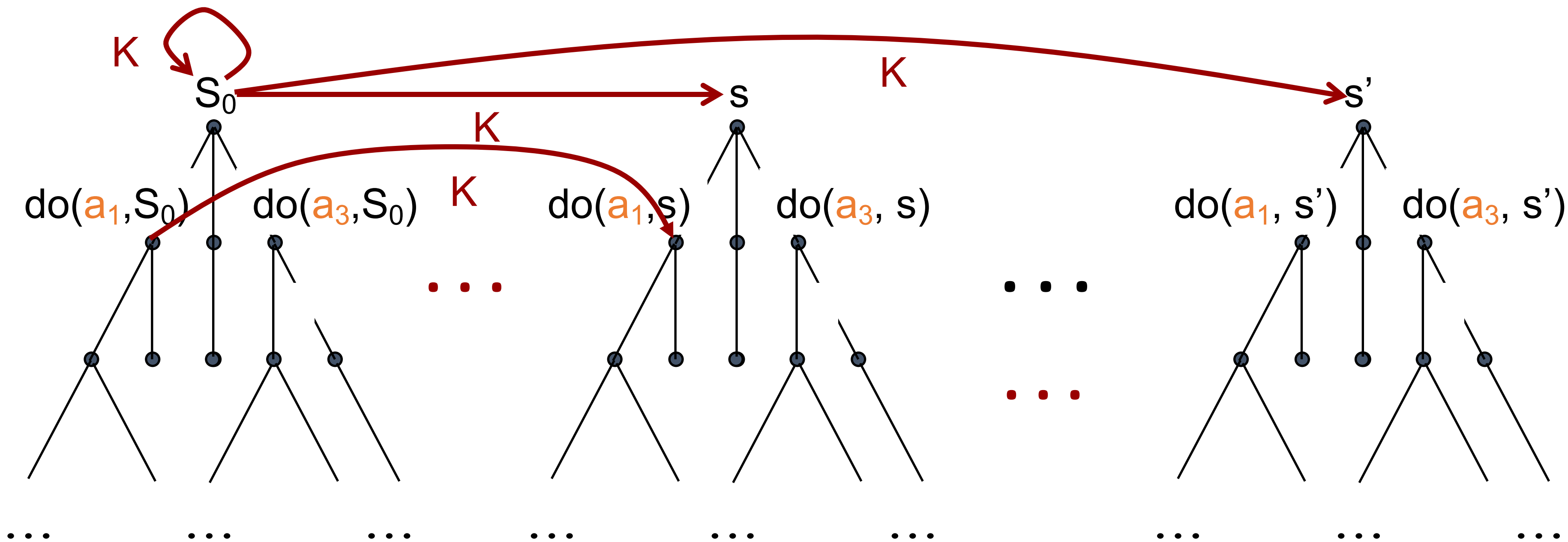
ok(power,s)
¬ plugged-in(radio,s)
ok(radio,s)
on(radio,s)
…

ok(power,s)
plugged-in(radio,s)
¬ ok(radio,s)
on(radio,s)
…



[Scherl & Levesque,93]
[Reiter 00]

Knowledge Fluent/Accessibility Relation K



Knowledge Fluent K(s',s)

Knows(ϕ,s) $\stackrel{def}{=}$ $\forall$s' K(s',s) $\supset$ ϕ(s')
Knows(on(radio),s) $\stackrel{def}{=}$ $\forall$s' K(s',s) $\supset$ on(radio,s')

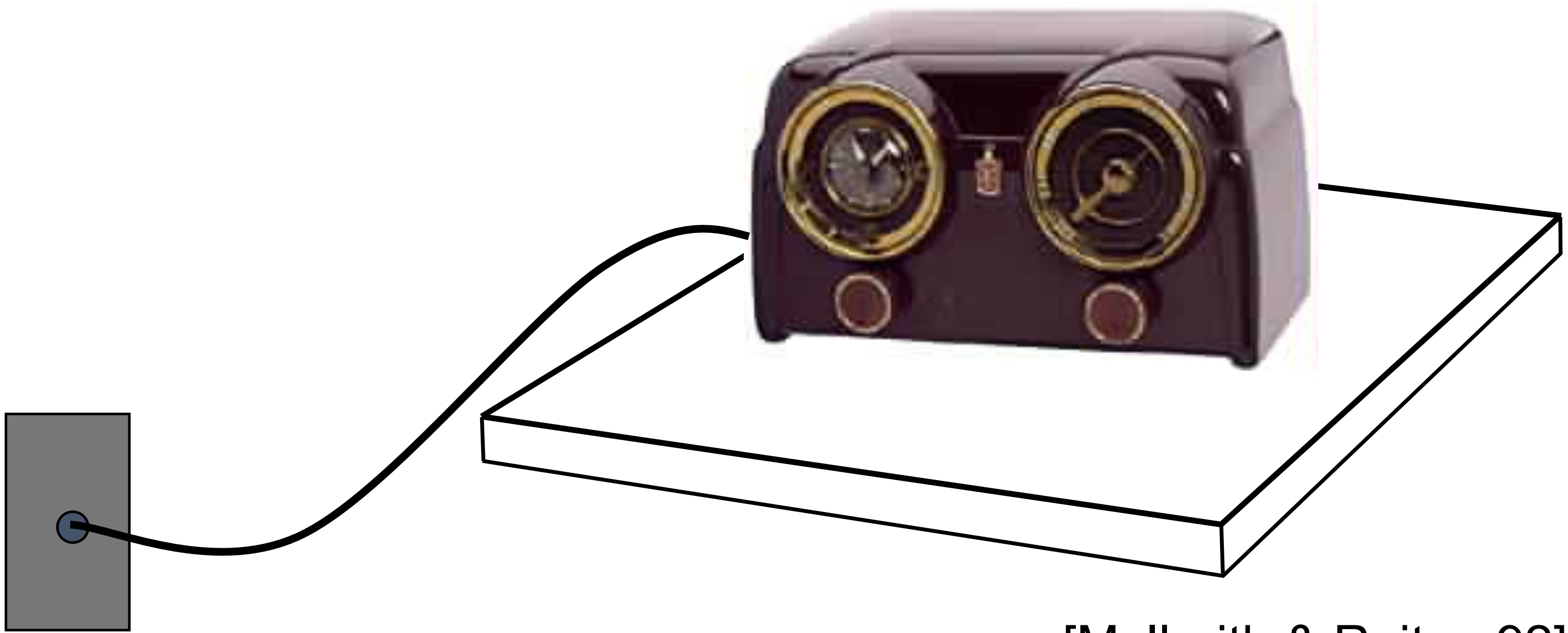
Kwhether(ϕ, s) $\stackrel{def}{=}$ **Knows**(ϕ,s) $\vee$ **Knows**($\neg$ ϕ,s)

Kwhether(on(radio), s) $\stackrel{def}{=}$ **Knows**(on(radio), s) $\vee$ **Knows**($\neg$ on(radio), s)

Definition of a Test

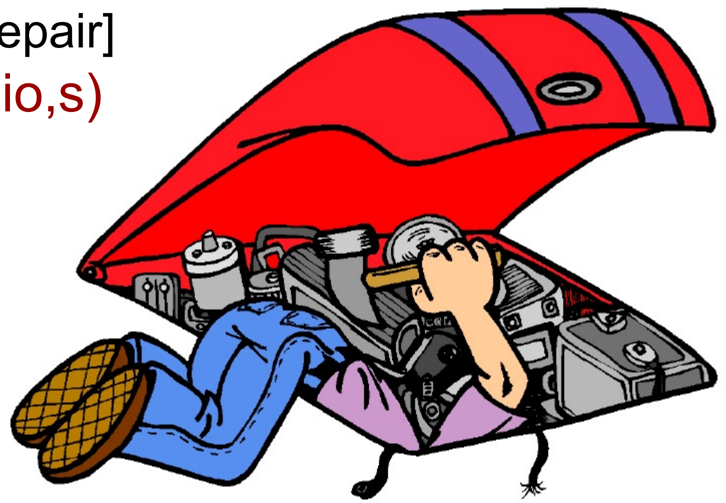
Simple Test:
A simple test is a pair (I,a) where I, the initial conditions, is a conjunction of literals, and a is a binary sense action.

E.g., (on(radio), listen(radio))



[McIlraith & Reiter, 92]
[McIlraith, 94]

Tests for Hypothesis Spaces



Car Domain Example [Idiots Guide to Car Repair]

(1) ab(battery,s) $\wedge$ on(radio,s) $\supset \neg$ noise(radio,s)
(2) ab(radio,s) $\supset \neg$ noise(radio,s)
(3) sparking $\supset$ sparks(s)
(4) sparks(s) $\wedge$ gas_leak(s) $\supset$ explosion(s)
(5) $\neg$ explosion(s)

. . .

Test of Hypothesis Space *HYP*:

A test (I,a) is a test for hypothesis space *HYP* in situation s iff
D $\wedge$ I $\wedge$ Poss(a,s) $\wedge$ H(s) is **satisfiable for every H $\in$ *HYP*.**

E.g.,

Hyp = {gas_leak(s), ab(battery,s), ab(spark_plugs,s), empty(tank,s)}
test (sparking, check_sparking(spark_plugs)) **is not a test** for hypothesis space *HYP*.

Confirmation and Refutation

Car Domain Example **(repeated)**

(1) $ab(battery,s) \wedge on(radio,s) \supset \neg\ noise(radio,s)$

(2) $ab(radio,s) \supset \neg\ noise(radio,s)$

. . .

Confirmation and Refutation:

The outcome α of test (I,a) **confirms** H $\in$ *HYP* iff

- $D \wedge I \wedge Poss(a,s) \models$ **Knows**$(H \supset \alpha, s)$

The outcome α of test (I,a) **refutes** H $\in$ *HYP* iff

- $D \wedge I \wedge Poss(a,s) \models$ **Knows**$(H \supset \neg\alpha, s)$

E.g.,

Hyp = {gas_leak(s), ab(battery,s), ab(spark_plugs,s), empty(tank,s)}
test = (on(radio), listen(radio))
outcome *noise(radio,s)* **refutes** hypothesis ab(battery,s).
outcome $\neg$*noise(radio,s)* **confirms** hypothesis ab(battery,s).

Discriminating Tests

<u>Discriminating Test:</u>

A test (I,a) is a **discriminating test** for hypothesis space *HYP* iff

- $D \wedge I \wedge Poss(a,s) \wedge H(s)$ is satisfiable for every $H \in HYP$, and
- There exists $H_i, H_j \in HYP$ such that outcome α of test (I,a) **refutes** either H_i or H_j no matter what the outcome.

If $H_i = \neg H_j$, (I,a) is an **individual discriminating test**.

E.g.,

Hyp = {gas_leak(s), ab(battery,s), ab(spark_plugs,s), empty(tank,s)}

test (true, check_empty(tank)) is an **individual discriminating test**.

<u>Other Tests:</u>

- relevant test
- constraining test

Testing Hypotheses

The efficacy of a test depends on the criteria defining the hypothesis space.

Consistency-based hypotheses

Given outcome α of test (I,a), any H $\in$ *HYP* such that

$D \wedge I \wedge Poss(a,s) \wedge H(s) \wedge \alpha$ is satisfiable.

Abductive hypotheses

Given outcome α of test (I,a), any H $\in$ *HYP* such that

$D \wedge I \wedge Poss(a,s) \wedge H(s) \models \alpha$.

Proposition (informally):

The outcome α of test (I,a) eliminates those consistency-based hypotheses H(s) $\in$ *HYP* that are **refuted** by test outcome α.

The outcome α of test (I,a) eliminates those abductive hypotheses H(s) $\in$ *HYP* that are **not confirmed** by test outcome α.

Testing Hypotheses

Proposition (informally):
Any outcome α of a relevant test (I,a) can eliminate abductive hypotheses.

Only a refuting outcome, α can eliminate a consistency-based hypotheses.

Discriminatory test outcomes (by defn) can eliminate either a consistency-based or an abductive hypothesis, regardless of the outcome.

[McIlraith & Reiter, 92]
[McIlraith, 94]

Testing Hypotheses

The efficacy of a test depends on the criteria defining the hypothesis space.

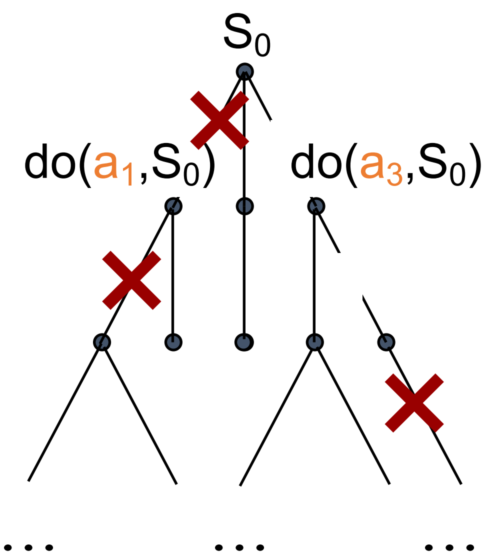
Proposition (informally):
Any outcome α of a relevant test (I,a) can eliminate abductive hypotheses.

Only a refuting outcome, α can eliminate a consistency-based hypotheses.

Discriminatory test outcomes (by defn) can eliminate either a consistency-based or an abductive hypothesis, regardless of the outcome.

[McIlraith & Reiter, 92]
[McIlraith, 94]

Complex Tests as Knowledge-Based Programs



<u>Golog [Levesque et al, 97]</u>

- sequencing
- if-then-else
- while-do
- nondeterministic choice

etc.

Proc CHECKBATTERY
TURN_ON(RADIO); LISTEN(RADIO);
if ¬ **Kwhether**(AB(BATTERY) **then**
(TURN_ON(LIGHTS); LOOK(LIGHTS)**)**;
if ¬ **Kwhether**(AB(BATTERY) **then**
(if ¬ **Kwhether**(AB(FUSES) **then** CHECKFUSES);
if Knows(¬ AB(FUSES) **then** METERCHECKBATTERY
else (FIXFUSES; CHECKBATTERY)**)**
endProc

OBSERVE: Complex tests can have side-effects on the world.

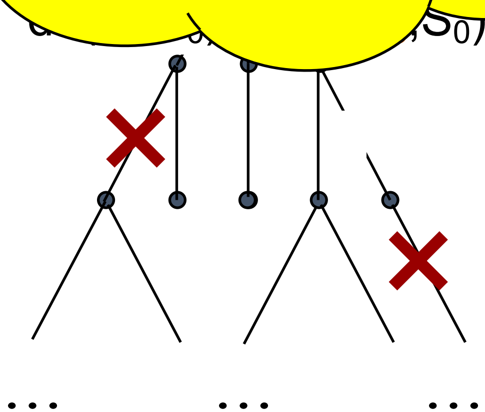
Complex Tests as Knowledge

conditional plans that can be used to realize diagnostic goals

Golog [Levesque et al, 97]

- sequencing
- if-then-else
- while-do
- nondeterministic choice

etc.



```
Proc CHECKBATTERY
   TURN_ON(RADIO); LISTEN(RADIO);
   if ¬ Kwhether(AB(BATTERY) then
          (TURN_ON(LIGHTS); LOOK(LIGHTS));
   if ¬ Kwhether(AB(BATTERY) then
          (if ¬ Kwhether(AB(FUSES) then CHECKFUSES);
           if Knows( ¬ AB(FUSES) then METERCHECKBATTERY
                                      else (FIXFUSES; CHECKBATTERY))
endProc
```

OBSERVE: Complex tests can have side-effects on the world.

Test Verification and Generation

Verification: We can automatically verify certain properties of a restricted class of complex tests, e.g.,

<u>Proving</u>	**<u>Verifies that the procedure</u>**
$\bigvee_{H \in HYP}$ **Kwhether**(H,s)	reduces the hypothesis space *HYP*
$\bigvee_{H \in HYP}$ **Knows**($\neg$ H,s)	is a discriminating test for *HYP*

<u>Theorem (informally stated):</u>
Regression rewriting reduces the verification problem to theorem proving in the initial situation.

Generation: We can automatically generate an even more restricted class of complex tests that satisfy particular properties, e.g.,

Kwhether(ab(battery),s)

in a brute-force manner by searching through the space of conditional plans, followed by regression and theorem proving in the initial situation. (not efficient!)

[Lesperance, 94]

And if I had time for one more thing …

- The utility of (Logical) Smoothing and Filtering in these behavioral interpretation tasks when we are dealing with partial observability [Mombourquette, Muise, M, AAAI17], [Amir, Russell, IJCAI03]

RECAP

about people

about machines/devices

about the world around us

Plan Recognition

Goal Recognition

Explanation Generation

Automated Diagnosis

Video Analysis

Behavior Interpretation

Activity Recognition

Intent Recognition

Narrative Understanding

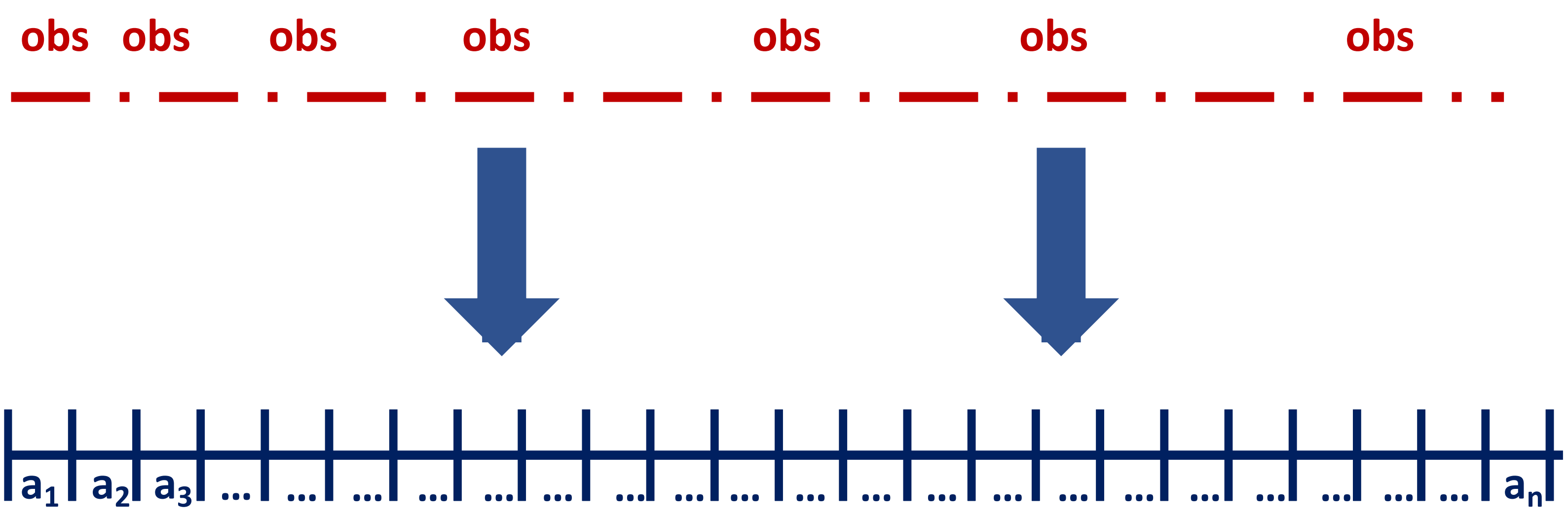
Auditing/Monitoring of Business Processes

Environmental Interpretation

Pattern of Inference

Align

- **observations** realized over time, with
- some **expectation of behaviour**



Relationship to Planning

Various tasks related to behavior interpretation can be realized by AI planning:

- Non-classical planning
- Conformant planning
- Conditional planning (i.e., offline contingent planning)
- Contingent planning
- Epistemic Planning

Continuing advances in planning technologies are enabling us to revisit and make progress on tasks that historically we understood how to specify but were unable to realize computationally. We see this in some of the work that follows.

Observations from the flashlight example

- Goal/Plan recognition can be done **post hoc or online**
- **Observations of action** *and* **state** are both relevant & useful
- **Goals** can be **temporally extended**
- **Goals** can be **epistemic**
- The actor (me) **used beliefs about the observers' (changing) models to realize her goal**
- The **observer can have agency** to **sense/reason/act** to expedite or make possible recognition or to assist or impede goal realization
- It's important to model the **actions** in the context of the **environment**
- The recognition task is often **purposeful** – you need not find a unique answer/solution. Often one need only discriminate sufficiently to **decide how to act**.

Can any of this inform goal recognition?

Stay Tuned

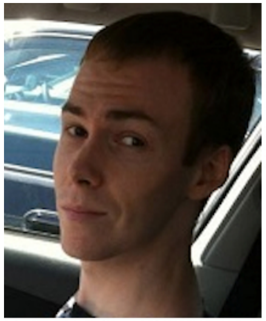
- **Active Goal Recognition** can be achieved by integrating ideas of sensing and test generation to construct contingent plans that can expedite or make possible goal recognition [Shvo, M, AAAI2020] (Spotlight and poster on Monday)
- **Epistemic Plan Recognition** Epistemics and theory of mind play a critical role in plan recognition [Shvo, Klassen, Sohrabi, M, PAIR2020 & AAMAS 2020] (Next talk)

Acknowledgements

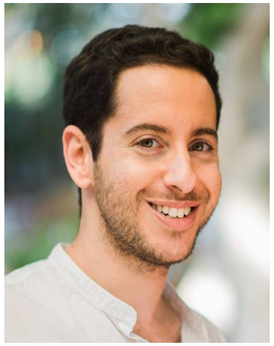
Jorge Baier

Shirin Sohrabi

Christian Fritz



Brent Mombourquette



Maayan Shvo

References (those mentioned)

Epistemic Plan Recognition. Shvo, M.; Klassen, T. Q.; Sohrabi, S.; and McIlraith, S. A. In *Proceedings of the 19th Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2020. To appear

Active Goal Recognition. Shvo, M.; and McIlraith, S. A. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Diagnostic Problem Solving via Planning with Ontic and Epistemic Goals. Baier, J. A.; Mombourquette, B.; and McIlraith, S. A. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2014.

Preferred Explanations: Theory and Generation via Planning. Sohrabi, S.; Baier, J. A.; and McIlraith, S. A. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI-11)*, pages 261-267, San Francisco, USA, August 2011.

Diagnosis as Planning Revisited. Sohrabi, S.; Baier, J. A.; and McIlraith, S. A. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR-10)*, pages 26-36, Toronto, Canada, May 2010. An abridged version of this paper appears in International Workshop on Principles of Diagnosis (DX-10).

What Sensing Tells Us: Towards a Formal Theory of Testing for Dynamical Systems. McIlraith, S.; and Scherl, R. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI'2000)*, pages 483-490, Austin, Texas, USA, July 30 - August 3 2000.

Explanatory Diagnosis: Conjecturing Actions to Explain Observations. McIlraith, S. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 167-179, Trento, Italy, June 2-5 1998.

Representing Actions and State Constraints in Model-Based Diagnosis. McIlraith, S. In *Proceedings Fourteenth National Conference on Artificial Intelligence (AAAI'97)*, pages 43-49, Providence, Rhode Island, USA, July 27-31 1997.

On Tests for Hypothetical Reasoning. McIlraith, S.; and Reiter, R. In Hamschers, W.; and Console, L., editor(s), *Readings in Model-Based Diagnosis*, pages 89-95. Morgan Kaufmann, 1992.

Logical Filtering and Smoothing: State Estimation in Partially Observable Domains. Mombourquette, B.; Muise, C.; and McIlraith, S. In *The 31st AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.

Etc.