Lecture Eleven: Diagnostic Agents

301315 Knowledge Representation and Reasoning
©Western Sydney University (Yan Zhang)

What Is a Diagnostic Agent

Recording the History of a Domain

- Recording the History of a Domain - Syntax
- Semantics
- Entailment

Briefcase Domain Revisited

Defining Explanations

- System Configurations
- Possible Explanations
- Diagnosing the Circuit

Computing Explanations

- The Problem
- Detecting a Symptom
- Finding Explanations

Tutorial and Lab Exercises

What Is a Diagnostic Agent

In a dynamic domain, we need to build an intelligent agent who is capable of finding explanations of unexpected observations. To achieve this goal, we need:

- a model of what is expected in the first place,
- a method of making and recording observations,
- and a method of detecting when reality doesn't match expectations.

What Is a Diagnostic Agent

- Previously, we were only interested in agent actions.
- Now we are also interested in modeling exogenous actions, which are those performed by nature or by other agents.
- Therefore, we will split our actions into these two types: **agent actions** and **exogenous actions**.

What Is a Diagnostic Agent

Two assumptions:

- The agent is capable of making correct observations, performing actions, and recording these observations and actions.
- *Normally* the agent is capable of observing all relevant exogenous actions occurring in its environment.

Note that the second assumption is defeasible.

What Is a Diagnostic Agent

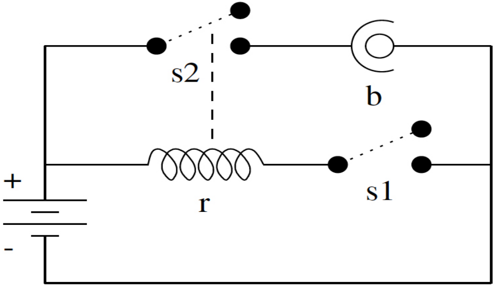
The Diagnostic Problem:

- A **symptom** consists of a recorded history of the system such that its last collection of observations is unexpected; i.e., it contradicts the agentss expectations.
- An **explanation** of a symptom is a collection of unobserved past occurrences of exogenous actions which may account for the unexpected observations.
- **Diagnostic Problem**: Given a description of a dynamic system and a symptom, find a possible explanation of the latter.

What Is a Diagnostic Agent

Example

Consider an agent controlling a simple electrical system:



It is aware of two exogenous actions: break (breaks bulb) and surge (breaks relay and breaks bulb if bulb unprotected).

What Is a Diagnostic Agent

Suppose initially:

- the bulb is protected
- the bulb is OK
- the relay is OK
- agent closes s_1

Agent expects the that the relay would become active causing s_2 to close and the bulb to emit light. What should it think if it observes that the light is not lit?

What Is a Diagnostic Agent

Possible explanations:

1. break occurred.
2. surge occurred.
3. break and surge occurred in parallel.

Humans tend to prefer minimal explanations.

- If the agent observes that the bulb is OK, then the only possible minimal explanation is surge.
- If the bulb was observed to be broken, then break is the explanation.
- If the bulb had not been protected, then both explanations would be valid.

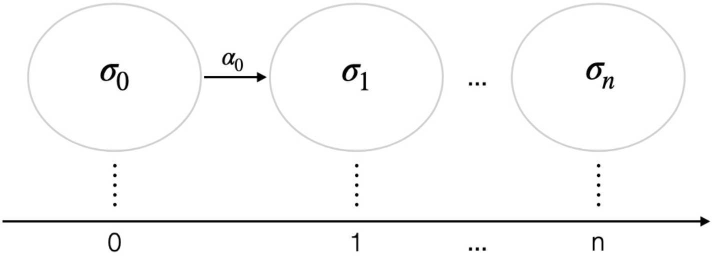
Recording the History of a Domain

In addition to a description of the transition diagram representing possible trajectories of the system, the knowledge base of a diagnostic agent contains the system's **recorded history** - observations made by the agent together with a record of its own actions

Recording the History of a Domain

- In order to reason about the past, the agent must have a record of the actions and observations it made.
- This recorded history defines a collection of paths that can be viewed as the system's possible pasts.
- Complete knowledge $=$ 1 path

Recording the History of a Domain - Syntax



Up to the current step n, at each step i ($i = 0, \cdots, n-1$), the agent *observes* somethings

Recording the History of a Domain - Syntax

The **recorded history** Γ_{n-1} of a system up to a current step n is a collection of **observations** that come in one of the following forms:

- $obs(t, true, i)$ - fluent f was observed to be true at step i; or
- $obs(t, false, i)$ - fluent f was observed to be false at step i; or
- $hpd(a, i)$ - action a was performed by the agent or observed to happen at step i,

where i is an integer from the interval $[0, n)$.

Recording the History of a Domain - Semantics

Semantics tells us how to match the set of *obs* and *hpd* statements with a transition diagram.

A path $M =< \sigma_0, a_o, \cdots, \sigma_{n-1}, a_{n-1}, \sigma_n >$ in the transition diagram $\mathcal{T}(\mathcal{SD})$ is a **model of a recorded history** Γ_{n-1} of dynamic system $\mathcal{SD}$ if for any $0 \leq i < n$:

(a) $a_i = \{a : hpd(a, i) \in \Gamma_{n-1}\}$;

(b) if $obs(f, true, i) \in \Gamma_{n-1}$, then $f \in \sigma_i$;

(c) if $obs(f, false, i) \in \Gamma_{n-1}$, then $\neg f \in \sigma_i$.

We say that Γ_{n-1} is **consistent** if it has a model.

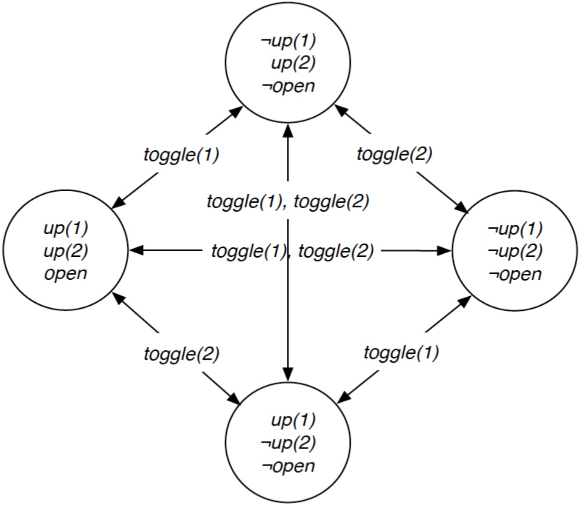
Recording the History of a Domain - Entailment

A fluent literal l, i.e., l is f or $\neg f$, **holds** in a model M of Γ_{n-1} at step i ($i \leq n$), denoted by $M \models holds(f, i)$, if $f \in \sigma_i$ (or $M \models \neg holds(f, i)$, if $\neg f \in \sigma_i$, resp.).

A recorded history Γ_{n-1} **entails** $holds(f, i)$ (or $\neg holds(f, i)$ if l is $\neg f$), denoted by $\Gamma_{n-1} \models holds(f, i)$ (or $\Gamma_{n-1} \models \neg holds(f, i)$, resp.), if for *every* model M of Γ_{n-1}, $M \models holds(f, i)$ (or $M \models \neg holds(f, i)$, resp.).

Briefcase Domain Revisited

Transition diagram for Briefcase domain:



Briefcase Domain Revisited

$toggle(C)$ **causes** $up(C)$ **if** $\neg up(C)$
$toggle(C)$ **causes** $\neg up(C)$ **if** $up(C)$
$open$ **if** $up(1), up(2)$

Suppose that, initially, clasp 1 was fastened and the agent unfastened it. The corresponding recorded history is:

$$\Gamma_0 = \left\{ \begin{array}{c} obs(up(1), false, 0) \\ hpd(toggle(1), 0) \end{array} \right.$$

What are the possible models of Γ_0 that satisfy this history?

Briefcase Domain Revisited

Γ_0 has two models:

$M_1 =$
$< \{\neg up(1), \neg up(2), \neg open\}, toggle(1), \{up(1), \neg up(2), \neg open\} >$

$M_2 =$
$< \{\neg up(1), up(2), \neg open\}, toggle(1), \{up(1), up(2), open\} >$

According the definition, we conclude that

$$\Gamma_0 \models \neg holds(up(1), 0),$$
$$\Gamma_0 \models \neg holds(open, 0),$$
$$\Gamma_0 \models holds(up(1), 1).$$

Briefcase Domain Revisited

An inconsistent history:

$$\Gamma_0 = \left\{ \begin{array}{c} obs(up(1), true, 0) \\ obs(up(2), true, 0) \\ hpd(toggle(1), 0) \\ obs(open, true, 1) \end{array} \right.$$

There is no path in our diagram that we can follow in this situation.

Briefcase Domain Revisited

10 min classroom exercise

Suppose the recorded history is

$$\Gamma_0 = \begin{cases} obs(up(1), true, 0) \\ hpd(toggle(2), 0) \end{cases}$$

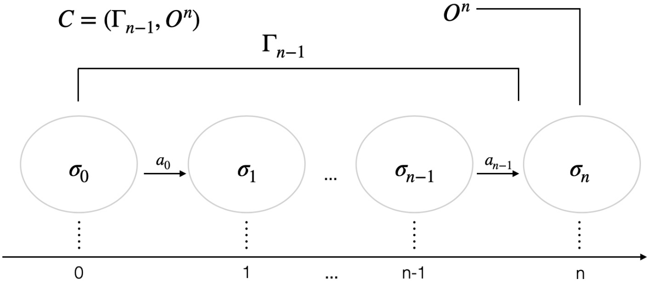
Find all models of Γ_0.

Defining Explanations - System Configurations

Now we consider an agent which completed the execution of its n^{th} action, and we denote the recorded history of the system up to this point by Γ_{n-1}.

- An agent just performed its nth action.
- The recorded history is Γ_{n-1}.
- The agent observes the values of fluents at step n, we call these observations O^n.
- The pair $\mathcal{C} = (\Gamma_{n-1}, O^n)$ is often referred to as the **system configuration**.

Defining Explanations - System Configurations



Defining Explanations - System Configurations

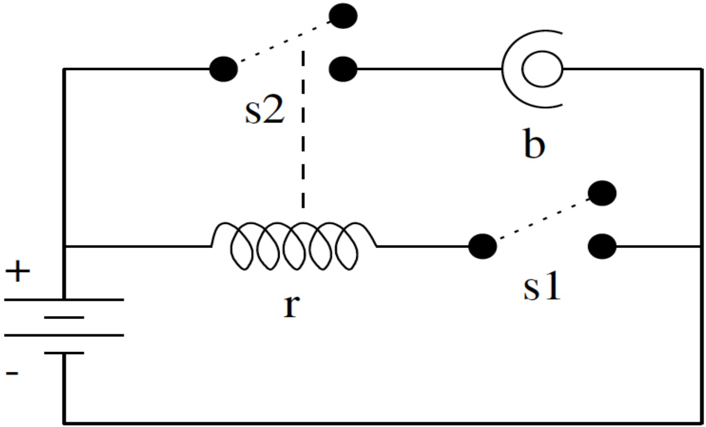
- If the new observations are consistent with the agent's view of the world (i.e., $\mathcal{C}$ is consistent), then the observations simply become part of the recorded history.
- Otherwise, it seeks an explanation which is that some exogenous action occurred that the agent did not observe.

Defining Explanations - Possible Explanations

Definition

- A configuration $\mathcal{C} = (\Gamma_{n-1}, O^n)$ is called a **symptom** if it is inconsistent, i.e., $\Gamma_{n-1} \cup O^n$ has no model.
- A **possible explanation** of a symptom $\mathcal{C}$ is a set $\mathcal{E}$ of statements $occurs(a, k)$ where a is an exogenous action, $0 \leq k < n$, and $\mathcal{C} \cup \mathcal{E}$ is consistent.

Defining Explanations - Diagnosing the Circuit



Defining Explanations - Diagnosing the Circuit

Signature: here I write them in *clingo* syntax.

```
%% Components:
    comp(r). comp(b).
%% Switches:
    switch(s1). switch(s2).

%% Fluents:
  fluent(inertial, prot(b)).
  fluent(inertial, closed(SW)) :- switch(SW).
  fluent(inertial,ab(X)) :- comp(X).

  fluent(defined, active(r)).
  fluent(defined,on(b)).
```

Defining Explanations - Diagnosing the Circuit

```
%% Actions:

  action(agent,close(s1)).
  action(exogenous,break).
  action(exogenous,surge).
```

Laws:
here I write them in $\mathcal{AL}$ language syntax
Part I: causal laws, state constraints and executability conditions:

$close(s1)$ **causes** *closed(s1)*
$active(r)$ **if** $closed(s1), \neg ab(r)$
$closed(s2)$ **if** $active(r)$
$on(b)$ **if** $closed(s2), \neg ab(b)$
impossible $close(s1)$ **if** $closed(s1)$

Defining Explanations - Diagnosing the Circuit

Part II: The information about the system's malfunctioning (exogenous actions) is given by:

$$\begin{array}{l} \textit{break}\ \textbf{causes}\ ab(b) \\ \textit{surge}\ \textbf{causes}\ ab(r) \\ \textit{surge}\ \textbf{causes}\ ab(b)\ \textbf{if}\ \neg prot(b) \end{array}$$

This completes our system description.

Defining Explanations - Diagnosing the Circuit

Now suppose the recorded history of the system is as follows:

$$\Gamma_0 = \left\{ \begin{array}{l} hpd(close(s1), 0) \\ obs(closed(s1), false, 0) \\ obs(closed(s2), false, 0) \\ obs(ab(b), false, 0) \\ obs(ab(r), false, 0) \\ obs(prot(b), true, 0) \end{array} \right.$$

Defining Explanations - Diagnosing the Circuit

Now we let σ_0 and σ_1 are as follows:

$$\sigma_0 = \{prot(b)\},$$
$$\sigma_1 = \{prot(b), closed(s1), active(r), closed(s2), on(b)\}$$

Then it is easy to check that $M =< \sigma_0, close(s1), \sigma_1 >$ is the only model of Γ_0. So we have

$$\Gamma_0 \models holds(on(b), 1),$$

which means that the agent expects the bulb to be lit after closing switch $s1$.

Defining Explanations - Diagnosing the Circuit

Now let us consider another scenario. Suppose that the agent observes that the bulb is not lit, i.e., its prediction differs from reality.

In our formal representation, it means that we have the following configuration:

$$\mathcal{C} = (\Gamma_0, \mathit{obs}(\mathit{on}(b), \mathit{false}, 1)),$$

which is a **symptom** - $\mathcal{C}$ has no model.

Defining Explanations - Diagnosing the Circuit

- This symptom may have three possible explanations:

$$\begin{aligned}\mathcal{E}_1 &= \{occurs(surge, 0)\},\\ \mathcal{E}_2 &= \{occurs(break, 0)\},\\ \mathcal{E}_3 &= \{occurs(surge, 0), occurs(break, 0)\}.\end{aligned}$$

- Actions break and surge are the only exogenous actions available in our language, and $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ are the only sets such that $\mathcal{C} \cup \mathcal{E}_i$ is consistent.

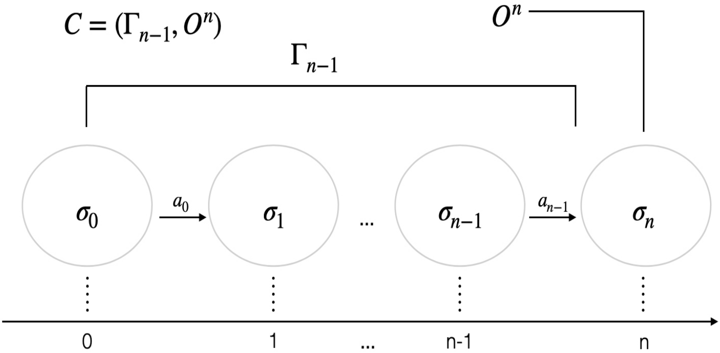
Computing Explanations - The Problem

Now let us consider a system with current configuration

$$\mathcal{C} = (\Gamma_{n-1}, O^n).$$

- The agent associated with the system just performed its n^{th} action and observed the values of some fluents.
- Now the agent needs to check that this configuration is consistent with the expectations, i.e., that $\mathcal{C}$ is not a symptom.
- If $\mathcal{C}$ is a symptom, then there should have a way to compute possible explanations.

Computing Explanations - The Problem



Computing Explanations - The Problem

- Recognize that there is a symptom.
- Consider possible, unobserved exogenous actions as explanations.

Computing Explanations - Detecting a Symptom

Questions:

(a) How to detect a symptom?

(b) How to compute an explanation?

Computing Explanations - Detecting a Symptom

We create an ASP program, called **all_clear**$(\mathcal{SD}, \mathcal{C})$, which consists of

- $\Pi(\mathcal{SD})$ - recall that we studied this encoding in lecture 9,
- the ASP encoding of $\mathcal{C}$, that is, the *clingo* syntax representation of $\Gamma_{n-1} \cup O^n$ - which is obviously,
- **plus** some extra rules ...

Computing Explanations - Detecting a Symptom

```
%% Full Awareness Axiom:
holds(F,0) | -holds(F,0) :- inertial_fluent(F).

%% Take what actually happened into account:
occurs(A,I) :- hpd(A,I).

%% Reality Check:
:- obs(F,true,I), -holds(F,I).
:- obs(F,false,I), holds(F,I).
```

If **all_clear**($\mathcal{SD}, \mathcal{C}$) has an answer set, which means it is consistent, then everything is OK. Otherwise, there is a symptom, and finding explanations is needed.

Computing Explanations - Detecting a Symptom

Proposition

A configuration $\mathcal{C}$ is a symptom if and only if (iff) program **all_clear**$(\mathcal{SD}, \mathcal{C})$ *is inconsistent, i.e., has no answer set.*

Computing Explanations - Finding Explanations

In order to find possible explanations, we create a program, called **diagnose**($\mathcal{SD}, \mathcal{C}$), which generates explanations.

Program **diagnose**($\mathcal{SD}, \mathcal{C}$) consists of

(a) program **all_clear**($\mathcal{SD}, \mathcal{C}$),

(b) **plus** some extra rules ...

Computing Explanations - Finding Explanations

```
%% The generator:
occurs(A,I) | -occurs(A,I) :- exogenous_action(A),
                              I < n, step(I).

%% This rule isolates actions that may be
%% part of an explanation:
expl(A,I) :- action(exogenous,A),
             occurs(A,I),  % Action A might have
                           % occurred
             not hpd(A,I). % Action A was not observed.
```

Tutorial and Lab Exercises

1. Consider the circuit domain studied in this lecture. Suppose we have the recorded history of the system as follows:

$$\Gamma_0 = \begin{cases} hpd(close(s1), 0) \\ obs(closed(s1), false, 0) \\ obs(closed(s2), false, 0) \\ obs(ab(b), false, 0) \\ obs(ab(r), false, 0) \\ obs(prot(b), false, 0) \end{cases}$$

Now the system has the configuration $\mathcal{C} = (\Gamma_0, obs(on(b), false, 1))$. Write a *clingo* program to check if $\mathcal{C}$ is a symptom or not. If yes, then extend your program and to generate possible explanations for this symptom.

Tutorial and Lab Exercises

2. In the Briefcase domain we discussed in this lecture, suppose there is an exogenous action given as follows:

   $$\mathit{loosen}(C)\ \textbf{causes}\ \ \neg \mathit{up}(C)\ \textbf{if}\ \mathit{up}(C)$$

   and a recorded history is:

   $$\Gamma_0 = \begin{cases} \mathit{obs}(\mathit{up}(1), \mathit{false}, 0) \\ \mathit{obs}(\mathit{up}(2), \mathit{true}, 0) \\ \mathit{hpd}(\mathit{toggle}(1), 0) \end{cases}$$

   Now the agent obtains a configuration $\mathcal{C} = (\Gamma_0, \mathit{obs}(\mathit{open}, \mathit{false}, 1))$. Write a *cling* program to check if $\mathcal{C}$ is a symptom, and if yes, extend your program to generate possible explanations.

Tutorial and Lab Exercises

3. Consider this scenario: *Mixing baking soda with lemon juice produces foam, unless the baking soda is stale. Joanna mixed baking soda with lemon juice, but there was no foam as a result.*
   (a) Using $\mathcal{AL}$ language to represent this story by a system description and a recorded history.
   (b) Translate this representation into ASP and check if the resulting configuration is a symptom. What knowledge does the agent have about the quality of Joanna's baking soda?