Abduction

Fariba Sadri

Deduction versus Abduction

Deduction: Given

A and

$A \rightarrow B$ you can infer

B.

Abduction: Given

B and

$A \rightarrow B$ you can abduce

A as an explanation for B.

Examples

Observation: *grass is wet*

You know:

it rained $\rightarrow$ *grass is wet*

Then there is a possible explanation: *it rained*

If you also know:

sprinkler was on $\rightarrow$ *grass is wet*

Then there are two possible explanations:

E_1: *it rained*

E_2: *sprinkler was on*

Example cntd.

If we have additional information such as:

cloudless sky

$cloudless\ sky \wedge it\ rained \rightarrow false$

i.e. $\neg\ (cloudless\ sky \wedge it\ rained)$

We can conclude $\neg\ (it\ rained)$

Then only one explanation is possible:

E_2: *sprinkler was on*

Informal Examples of Abductive Reasoning

- Medical diagnosis: given a set of symptoms, what is the diagnosis that would best explain most of them?
- Jury decision in a criminal case? Maybe. The Jury must consider whether the prosecution or the defence has the best explanation to cover all the points of evidence.

Applications of Abduction

Artificial Intelligence, in particular:

- Automatic planning
- Fault diagnosis
- Learning: Course “Logic-Based Learning” in the Spring term
- Default reasoning
- Belief revision
- .....

Deduction versus Abduction

Deductive reasoning:

We are given a Theory (Premises) and a Goal and we check if

$$\text{Theory} \vdash \text{Goal}$$

- Conclusion is "guaranteed"
- If the premises are true, then the conclusion must also be true

Deduction versus Abduction cntd.

Abductive reasoning:

We are given a Theory and an Observation and we want to find an explanation (maybe the “best” explanation) for the Observation.

- Hypothetical (not guaranteed)
- The explantion is our best shot
- Defeasible (new information can discredit an explanation)
- Described by Charles Sanders Peirce as "guessing"

Abduction can be Formalised in Logic

Informally:

Given a Theory and an Observation an abductive Explanation of the Observation is such that:

$$\text{Theory} \cup \text{Explanation} \vdash \text{Observation}$$

and Explanation is made up of "Abducible" predicates. Typically, abducibles have no definitions in the Theory.

In addition Explanation may have to satisfy some other properties.

Example

Theory: *it rained* $\rightarrow$ *grass is wet*

sprinkler was on $\rightarrow$ *grass is wet*

Observation: *grass is wet*

Abducibles: *it rained, sprinkler was on*

Explanations (Abductive Explanations):

E_1: *it rained*

E_2: *sprinkler was on*

$$\text{Theory} \cup E_1 \vdash \text{Observation}$$

$$\text{Theory} \cup E_2 \vdash \text{Observation}$$

Another Example

Theory: $a \wedge b \wedge e \rightarrow g$ $\quad a \quad c$

$c \wedge d \rightarrow g \quad c \rightarrow h$

$a \wedge f \wedge h \rightarrow g \quad e \rightarrow f$

Observation: g

Abducibles: b, e

Explanations (Abductive Explanations):

E_1: $b \wedge e$

E_2: e

Theory $\cup\ E_1 \vdash$ Observation

Theory $\cup\ E_2 \vdash$ Observation

How To Find an Abducible Solution

In Prolog Notation:

Theory: g :- $a \wedge b \wedge e$. a. c.

g :- $c \wedge d$. h :- c.

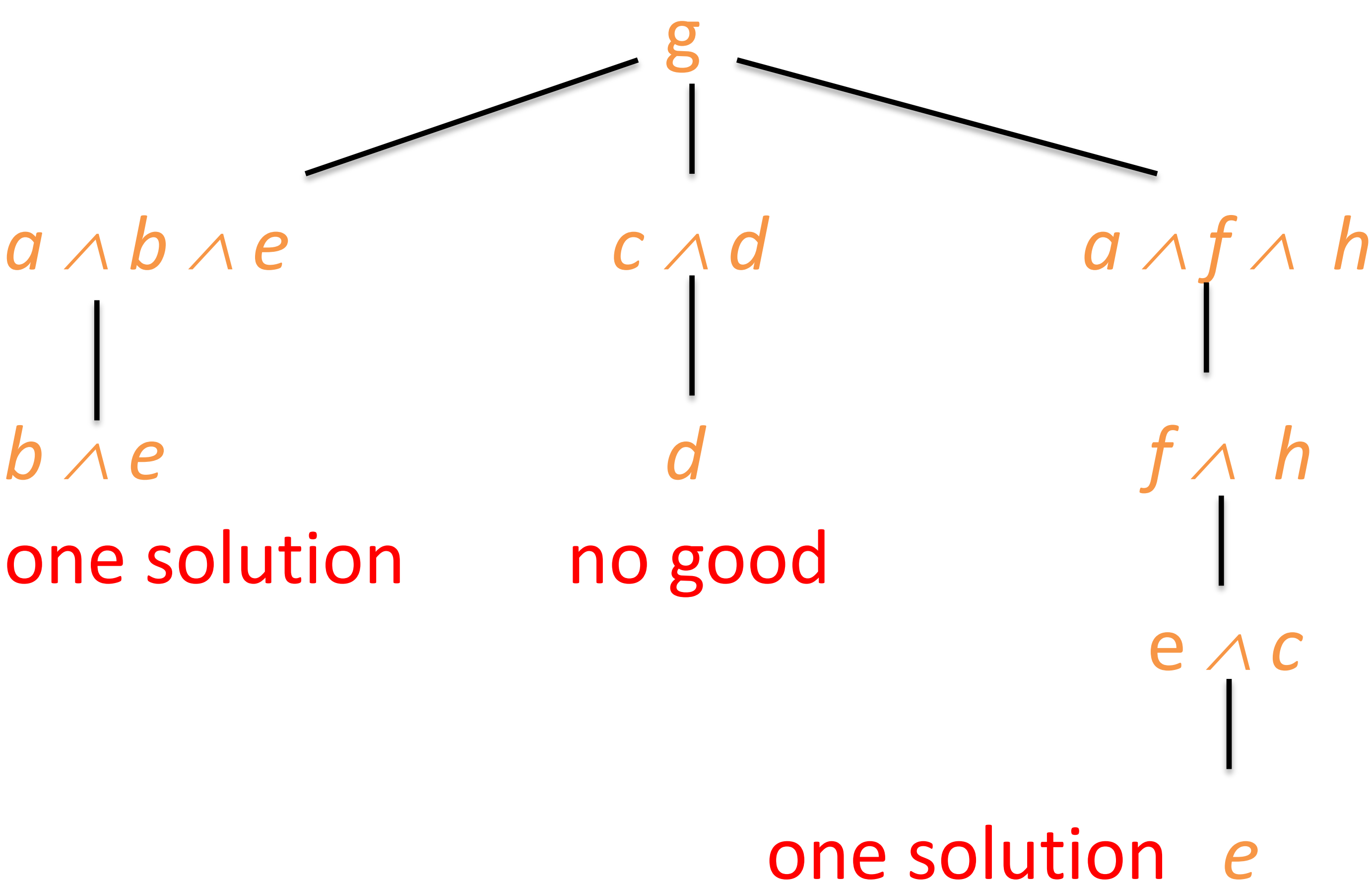
g :- $a \wedge f \wedge h$. f :- e.

Observation: g.

Abducibles: b, e

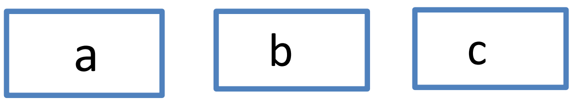
g :- a ∧ b ∧ e. ***g :- c ∧ d.*** ***g :- a ∧ f ∧ h.***

a ***c*** ***h :- c.*** ***f :- e.***

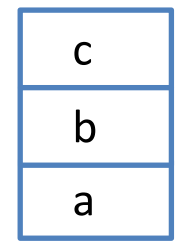


A Very Simple Example in Automatic Planning

Initial state



Goal state



Goal: $on(b,a) \wedge on(c,b)$

Theory: $\forall X,Y\ (move(X,Y) \rightarrow on(X,Y))$

or in Prolog: on(X,Y) :- move(X,Y).

Abducibles: all instances of move/2

Abductive solution: {move(b,a), move(c,b)}

Another Simple Example in Automatic Planning (using Prolog syntax)

```
prepare_for_trip :-      have_valid_passport ,
                         book_transport.
have_valid_passport :- \+ passport_expired.
have_valid_passport :- get_passport_renewed.
book_transport :- book_train.
book_transport :- book_plane.
passport_expired.
```

Goal: prepare_for_trip.

Abducibles:

{ get_passport_renewed, book_train, book_plane }

Two abductive solutions:

E_1= {get_passport_renewed, book_train}

E_2= { get_passport_renewed, book_plane}

	Observation O	Abducibles	Explanation E
Planning	Goal	Actions (Temporal Constraints)	Plan, e.g. set of actions
Diagnosis	Symptoms	Diseases/ Faults	Disease/ Fault

An Example from Diagnosis

http://web.stanford.edu/class/cs227/Lectures/lec12.pdf

Theory has facts and rules about symptoms and diseases.

Disease $\wedge$ Other_conditions $\rightarrow$ Symptoms

Goal: Hypothesise about diseases that best explain the symptoms.

Typically we won't have information such as

$$\text{Symptoms} \wedge \text{Other\_conditions} \rightarrow \text{Disease}$$

So reasoning is not deductive.

$tennis\_elbow \rightarrow sore\_elbow$

$arthritis \wedge untreated \rightarrow sore\_joints$

$sore\_joints \rightarrow sore\_elbow$

$sore\_joints \rightarrow sore\_hips$

$arthritis \wedge untreated \rightarrow problem\_with\_balance$

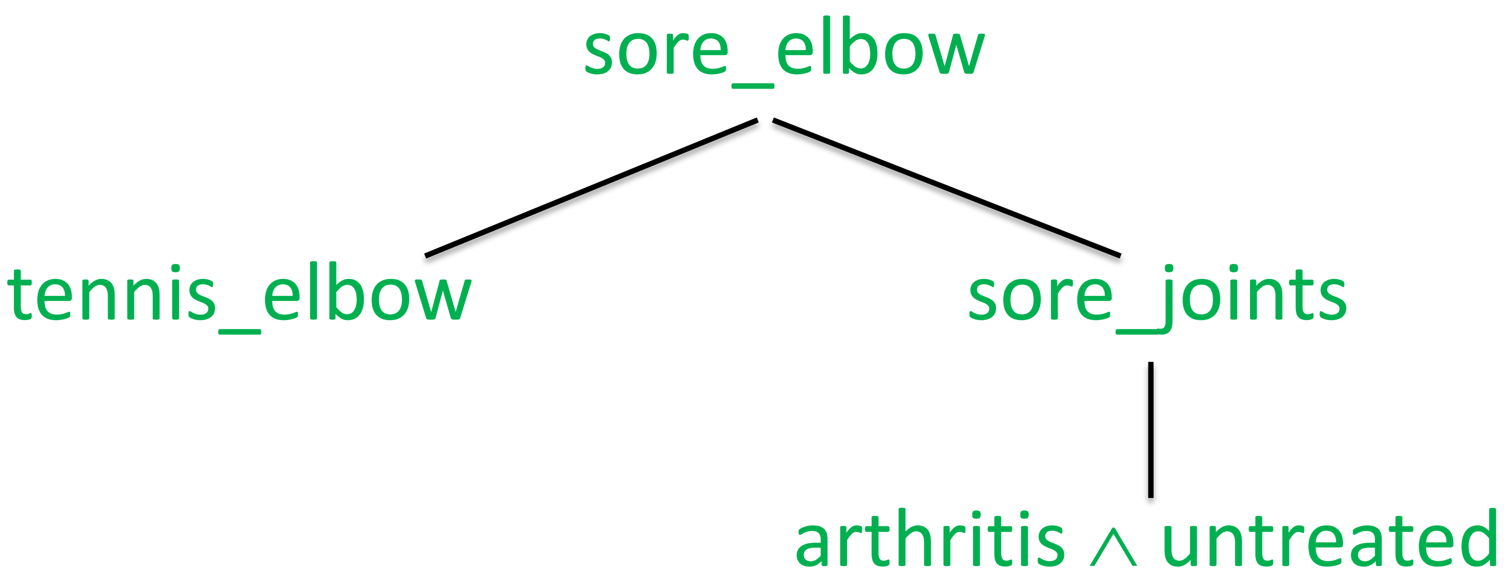
Symptom:

sore_elbow

Explanations :

tennis_elbow

arthritis, untreated


sore_elbow
tennis_elbow
sore_joints
arthritis ∧ untreated

Besides accounting for all of the known symptoms, a hypothesis may also predict other symptoms which have not yet been observed.

These predictions provide a way to test the validity of a hypothesis by making the observation and seeing if the result is consistent with the prediction.

A formal characterisation of abductive logic programs: Preliminaries

Definition: Logic Programs

Logic programs are sets of clauses of the form:

$$H \leftarrow A_1 \wedge \ldots \wedge A_n$$

In Prolog

$$H \text{ :- } A_1, \ldots, A_n.$$

H and all the A_i are atomic formulas. (In general the A_i can also be negative literals, but we will ignore this here.)

A formal characterisation of abduction abductive logic programs: Preliminaries ctd.

Definition: Integrity Constraints

Integrity Constraints are sentences of the form:

$A_1, \ldots, A_n \rightarrow$ false.

or equivalently

$\neg(A_1, \ldots, A_n)$.

where the A_i are atoms.

A formal characterisation of abductive logic programs

T is a **logic program**

H is a set of ***undefined*** ground **atoms** (the abducibles)

O is a (implicitly existentially quantified) conjunction of atoms (the goal/observations)

I is a set of integrity constraints

E (the abductive answer) is such that

$\mathbf{T} \cup \mathbf{E} \vdash O$

$E \subseteq H$

$\mathbf{T} \cup \mathbf{E}$ satisfies **I** $\qquad$ $\mathbf{T} \cup \mathbf{E} \cup \mathbf{I}$ consistent

Example

T: *it rained* $\rightarrow$ *grass is wet*
sprinkler was on $\rightarrow$ *grass is wet*
cloudless sky

H: *it rained, sprinkler was on*

O: *grass is wet*

I: *cloudless sky* $\wedge$ *it rained* $\rightarrow$ *false*
i.e. $\neg$ *(cloudless sky* $\wedge$ *it rained)*

E: *sprinkler was on*

Extending with negation (as failure)

Abductive logic programming works for *normal logic programs*, i.e. logic programs and integrity constraints that can have negative as well as positive conditions.

$H \leftarrow A_1 \wedge \ldots \wedge A_n \wedge \text{not } B_1 \wedge \ldots \wedge \text{not } B_n.$

$H \text{ :- } A_1 \wedge \ldots \wedge A_n \wedge \backslash + B_1 \wedge \ldots \wedge \backslash + B_n.$

- H and all the A_i and B_i are atomic formulas.
- The negation in the negative literals not B_i is referred to as "negation as failure"
- A negative condition not B_i is shown to hold by showing that the positive condition B_i fails to hold.

Integrity Constraints with Negation

$B_1, \ldots, B_n \rightarrow \text{false}.$

or equivalently

$\neg(B_1, \ldots, B_n).$

where the B_i are literals.

Abductive logic programs

T: a ***normal*** **logic program**

H: a set of ***undefined*** ground **atoms (abducible atoms)**

O: a (implicitly existentially quantified) conjunction of **literals** (atoms or NAF of atoms)

I: is a set of integrity constraints

E: (the abductive answer) is such that

$\mathbf{T} \cup \mathbf{E} \vdash_{\text{NAF}} \mathrm{O}$

$\mathrm{E} \subseteq \mathrm{H}$

$\mathbf{T} \cup \mathbf{E}$ satisfies $\mathbf{I}$

Example

```
T:      g(T) :- a1(T), \+p(T), a2(T).
        g(T) :- a1(T), a3(T).
        g(T) :- a4(T).
        p(10).
        q(T) :- a5(T).
I:       a4(T), \+ q(T) -> false.
        a3(T), \+ a6(T) -> false.
Abducibles:  a1, a2, a3, a4, a5, a6
```

Goal: g(10).

Abductive Answers:

E1: a1(10), a3(10), a6(10).

E2: a4(10), a5(10).

Another Example

P: have(X) :- buy(X).
have(X) :- borrow(X).
have(money).

A: buy, borrow, register (actions)

IC: buy(X), \+ have(money) $\rightarrow$ false
buy(tv), \+ register(tv) $\rightarrow$ false
buy(money) $\rightarrow$ false

Goal: have(tv)

$\Delta 1$: buy(tv), register(tv) (Plan 1)

$\Delta 2$: borrow(tv) (Plan 2)

P: have(X) :- buy(X).
have(X) :- borrow(X).

A: buy, borrow, register (actions)

IC: buy(X), \+ have(money) $\rightarrow$ false
buy(tv), \+ register(tv) $\rightarrow$ false
buy(money) $\rightarrow$ false

Goal: have(tv)

Δ1: borrow(money), buy(tv), register(tv) (Plan 1)

Δ2: borrow(tv) (Plan 2)