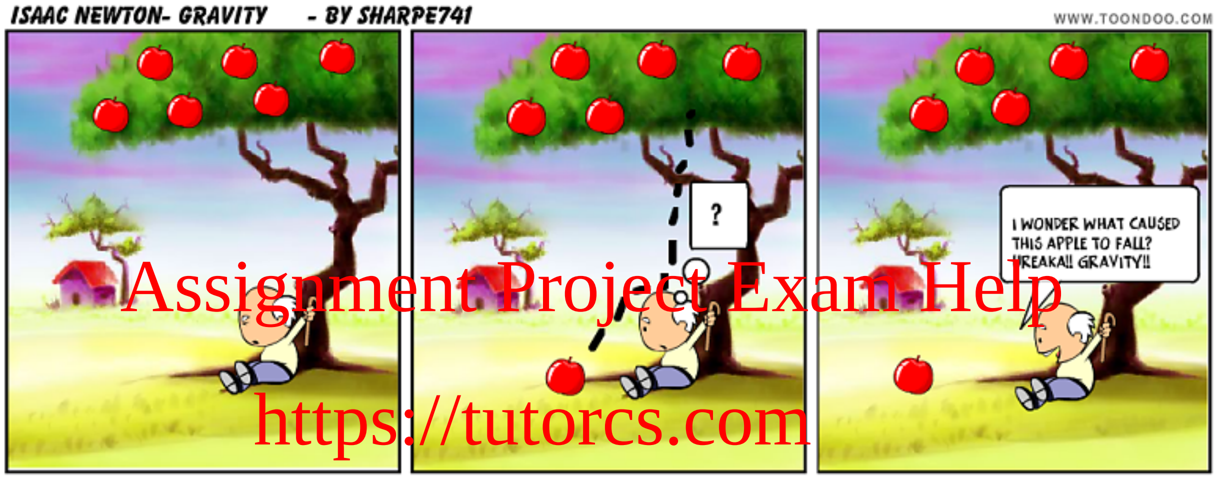
Abductive Inference

- Informal definition
- Formalizing the task
- Algorithm
- Semantic properties
- Example applications
  » Diagnosis problems
  » Automated Planning

Abductive Inference



When Newton saw the apple falling down, he must have done an abductive inference and came up with the theory of gravity.

- *Apple fell down.*
- *If earth pulled everything towards it, then of course, apple too would fall down.*
- *So earth is pulling everything towards it.*

Handling incomplete Information

"*If I push the switch button, the light in my room will switch on*"

Default reasoning:

reasoning about "normal circumstances", by making assumptions on what is false.

"*The light does not switch on! The lightbulb must be broken*"

Abductive reasoning:

reasoning about possible explanations, making assumptions on what might be false and might be true.

Given a *theory* and an *observation*, find an *explanation* such that

$$\textit{theory} \cup \textit{explanation} \vDash \textit{observation}$$

Desirable properties of explanations

theory

flies(X) ← bird(X), not abnormal(X)
abnormal(X) ← penguin(X)
bird(X) ← penguin(X)
bird(X) ← sparrow(X)

observation flies(tweety)

- Explanations should be **basic**

E1: {sparrow(tweety)}

basic explanation

E2: {bird(tweety)}

non basic explanation

They are restricted to **abducibles**, ground literals with predicates that are not defined in the theory.

Desirable properties of explanations

theory

- flies(X) ← bird(X), not abnormal(X)
- abnormal(X) ← penguin(X)
- bird(X) ← penguin(X)
- bird(X) ← sparrow(X)
- bird(X) ← woodpecker(X)

observation flies(tweety)

- Explanations should be **minimal**

E1: {sparrow(tweety)}

E2: {sparrow(tweety), woodpecker(tweety)}

minimal explanation	non minimal explanation

Should **not be subsumed** by any other explanation.

Desirable properties of explanations

theory

flies(X) ← bird(X), not abnormal(X)
abnormal(X) ← penguin(X)
bird(X) ← penguin(X)
bird(X) ← sparrow(X)
bird(X) ← woodpecker(X)
← woodpecker(tweety)

observation flies(tweety)

- Explanations should be **consistent with the theory**

E1: {sparrow(tweety)}

consistent explanation

E2: {woodpecker(tweety)}

non consistent explanation

Theory ∪ *E2* is **inconsistent**

Defining abductive reasoning

theory

flies(X) ← bird(X), not abnormal(X)
abnormal(X) ← penguin(X)
bird(X) ← penguin(X)
bird(X) ← sparrow(X)
bird(X) ← woodpecker(X)

constraints ← woodpecker(tweety)

observation flies(tweety)

Given a theory, BK, a set of integrity constraints IC, a set of abducibles A, abductive reasoning framework is the tuple <BK, A, IC>, where A includes ground literals whose predicate names are not defined in BK.

A = {sparrow(tweety), penguin(tweety), woodpecker(tweety)}

Formal definition of abduction

An abductive logic program, for a given problem domain, is:

$$\langle KB, A, IC \rangle \quad \begin{cases} KB = \text{set of normal clauses} \\ A = \text{set of ground undefined literals} \\ IC = \text{set of normal denials} \end{cases}$$

Given an abductive framework, an abductive solution, called explanation, for a given goal G, is a set Δ of ground literals such that:

- $\Delta \subseteq A$ belong to the predefined language of abducibles
- $KB \cup \Delta \models G$ provide missing information needed to solve the goal
- $KB \cup \Delta \not\models \perp$ is consistent with the knowledge base
- $KB \cup \Delta \models IC$ it satisfies the integrity constraints

Abduction: extending SLD

What happens when our knowledge base is incomplete?

KB

likes(peter, S) ← studentOf(S, peter)
likes(X, Y) ← friend(Y, X)

$\models$ likes(peter, paul)

← likes(peter, paul)

← studentOf(paul, peter) ?

← friend(paul, peter) ?

SLD would fail, due to lack of information.

We could instead assume (as possible explanations) what is not known.

Multiple equally good explanations:

Δ_1 = {studentOf(paul, peter)}

Δ_2 = {friend(paul, peter}

Abductive reasoning computes explanations of observations with respect to given KB

Abduction: what about NAF?

How do we guarantee consistency with KB when we have NAF?

KB

flies(X) ← bird(X), not abnormal(X)
abnormal(X) ← penguin(X)
bird(X) ← penguin(X)
bird(X) ← sparrow(X)

$\models$ flies(tweety)

← flies(tweety)

← bird(tweety), not abnormal(tweety)

← penguin(tweety), not abnormal(tweety) ?

← sparrow(tweety), not abnormal(tweety) ?

Multiple explanations:

Δ_1 = {penguin(tweety), not abnormal(tweety)}

Δ_2 = {sparrow(tweety), not abnormal(tweety)}

- Δ_1 is inconsistent with KB
- *not abnormal* is not abducible

→ We need to reason with *not* explicitly

Abduction: explaining NAF

How to maintain consistency when negated literals are assumed?

KB

flies(X) ← not abnormal(X), bird(X)
abnormal(X) ← penguin(X)
bird(X) ← penguin(X)
bird(X) ← sparrow(X)

⊨ flies(tweety)

← flies(tweety)

← not abnormal(tweety), bird(X)

← not abnormal(tweety) ?

Δ_1 = {not abnormal(tweety)}

← abnormal(tweety)

← penguin(tweety)

← not penguin(tweety) ?

□

Explicit explanation for *not abnormal(tweety)*

Δ_1 = {not abnormal(tweety), not penguin(tweety)}

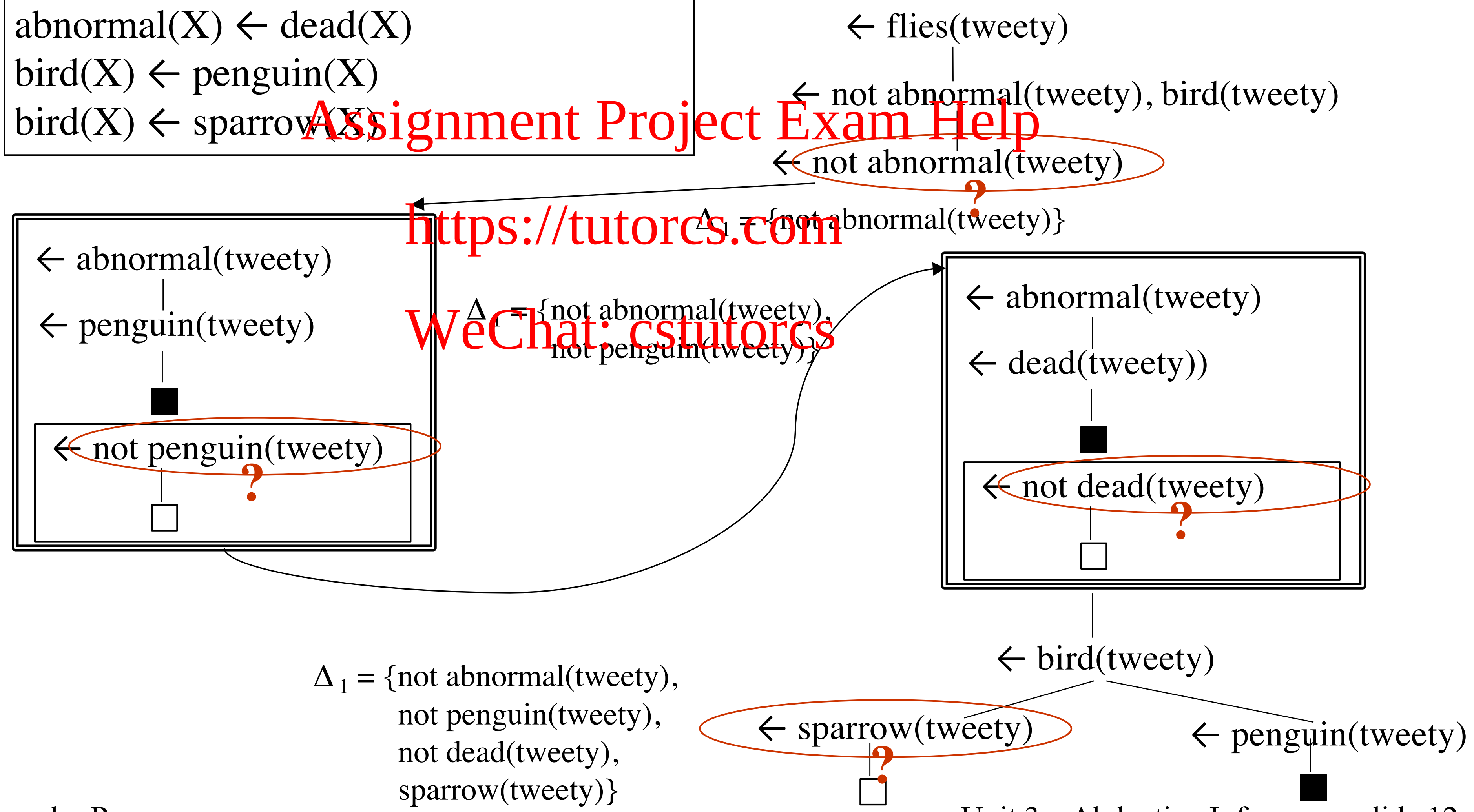
Abduction: explaining NAF

Consider the following example:

KB

flies(X) ← not abnormal(X), bird(X)
abnormal(X) ← penguin(X)
abnormal(X) ← dead(X)
bird(X) ← penguin(X)
bird(X) ← sparrow(X)

⊨ flies(tweety)



Abduction: explaining NAF

The order in which positive and negated literals appear in a clause only influences the order in which abducibles are added to the explanation, but not the explanation itself.

? flies(tweety)

flies(X) ← not abnormal(X), bird(X)
abnormal(X) ← penguin(X)
abnormal(X) ← dead(X)
bird(X) ← penguin(X)
bird(X) ← sparrow(X)

Δ_1 = { not abnormal(tweety),
not penguin(tweety),
not dead(tweety),
sparrow(tweety)}

flies(X) ← bird(X), not abnormal(X)
abnormal(X) ← penguin(X)
abnormal(X) ← dead(X)
bird(X) ← penguin(X)
bird(X) ← sparrow(X)

Δ_1 = { sparrow(tweety),
not abnormal(tweety),
not penguin(tweety),
not dead(tweety)}

Abduction: satisfying constraints?

Consider the following example:

KB	Ab	G
headache(X) ← overworked(X) headache(X) ← migrane(X) migrane(X) ← wrongdiet(X) migrane(X) ← jetlag(X)	overworked(jane) wrongdiet(jane) jetlag(jane)	headache(jane)

$\Delta_1 = \{overworked(jane)\}$

$\Delta_2 = \{wrongdiet(jane)\}$

$\Delta_3 = \{jetlag(jane)\}$

Alternative explanations

Abduction: satisfying constraints?

Consider the following example:

KB

headache(X) ← overworked(X)
headache(X) ← migrane(X)
migrane(X) ← wrongdiet(X)
migrane(X) ← jetlag(X)
student(jane)
← student(X), overworked(X)

Ab

overworked(jane)
wrongdiet(jane)
jetlag(jane)

G

headache(jane)

$\cancel{\Delta_1 = \{\text{overworked(jane)}\}}$

$\Delta_2 = \{\text{wrongdiet(jane)}\}$

$\Delta_3 = \{\text{jetlag(jane)}\}$

Constraints may eliminate explanations

Abduction: satisfying constraints?

KB

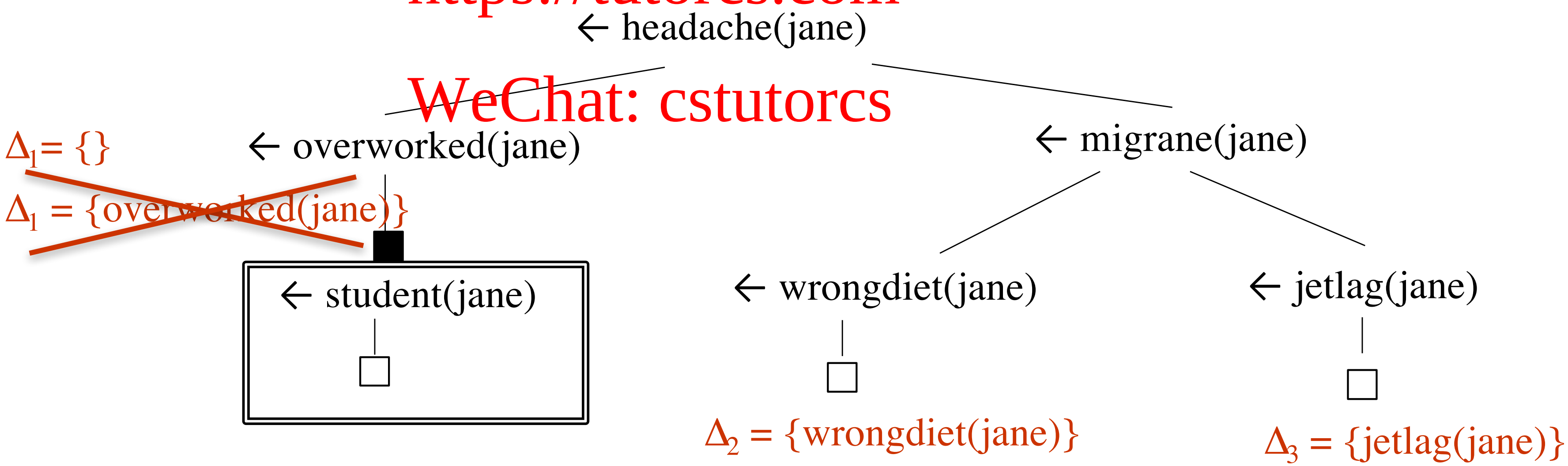
headache(X) ← overworked(X)
headache(X) ← migrane(X)
migrane(X) ← wrongdiet(X)
migrane(X) ← jetlag(X)
student(jane)
← student(X), overworked(X)

Ab

overworked(jane)
wrongdiet(jane)
jetlag(jane)

G

headache(jane)



Abduction: satisfying constraints?

Consider the following example:

KB	Ab	G
headache(X) ← overworked(X) headache(X) ← migrane(X) migrane(X) ← wrongdiet(X) ← not jetlag(X), overworked(X)	overworked(jane) wrongdiet(jane) jetlag(jane)	headache(jane)

$$\Delta_1 = \{overworked(jane), jetlag(jane)\}$$

$$\Delta_2 = \{wrongdiet(jane)\}$$

Constraints may force abducibles in explanations

Abduction: satisfying constraints?

KB

headache(X) ← overworked(X)
headache(X) ← migrane(X)
migrane(X) ← wrongdiet(X)
← not jetlag(jane)), overworked(X)

Ab

overworked(jane)
wrongdiet(jane)
jetlag(jane)

G

headache(jane)

← headache(jane)

← overworked(jane)

Δ_1 = {overworked(jane)}

← not jetlag(jane)

← jetlag(jane)

Δ_1 = {overworked(jane), jetlag(jane)}

← migrane(jane)

← wrongdiet(jaen)

Δ_2 = {wrongdiet(jane)}

Abductive proof procedure

Two reasoning phases:

- Abductive derivation: it proceeds similarly to SLDNF resolution, but with the additional feature of assuming abducibles where, encountered as sub-goals to be proved.

- Consistency derivation: resolves the assumed literal with all relevant integrity constraints and prove that each of the resolvants fails (possibly adding more assumptions if needed).

Note:

1. All negated literals are considered to be abducibles.
2. IC implicitly contains $\leftarrow$ P, not P (for every predicate P)

Abductive Proof Procedure

Let <KB, A, IC> be an abductive model expressed in normal clausal logic and let G be a ground observation:

Abductive phase

(G_1, Δ_1)

(G_2, Δ_2)

$\vdots$

$(\varnothing, \Delta_n)$

$G_1 = G$, initially $\Delta = \{\}$

Select a subgoal L from G_i; let $G_i' = G_i - \{L\}$

A1. $L \notin A$ and L is a positive atom

if $H \leftarrow B$ in KB such that $L = H\theta$

$G_{i+1} = B\theta \cup G_i'$ and $\Delta_{i+1} = \Delta_i$

A2. $L \in \Delta_i$ then $G_{i+1} = G_i'$ and $\Delta_{i+1} = \Delta_i$

A3. $L \in A$ and $L \notin \Delta_i$ and *not* $L \notin \Delta_I$, then

if there is a successful consistency derivation with $\Delta_i \cup \{L\}$, then $G_{i+1} = G_i'$ and $\Delta_{i+1} = \Delta'$

$(\Delta_{i+1} = \Delta')$

$(L, \Delta_i \cup \{L\})$

$\vdots$

$(\varnothing, \Delta')$

Consistency phase

Abductive Proof Procedure

Let <KB, A, IC> be an abductive model expressed in normal clausal logic and let G be a ground observation:

Consistency phase

$(B, \Delta_i \cup \{B\})$

(F_1, Δ_1)

$\vdots$

(F_n, Δ_n)

F_1 all denials in IC resolved with B

Select a denial $\leftarrow \phi_i$ in F_1 and a literal L from it;
let $\phi_i' = \phi_i - \{L\}$ and $F_1' = F_1 - \{\leftarrow \phi_i\}$.

C1. $L \notin A$, perform SLDNF failure with L as a subgoal

C2. $L \in \Delta_i$, then $F_{i+1} = \{\leftarrow \phi_i'\} \cup F_i'$ and $\Delta_{i+1} = \Delta_i$.

C3. $L \in A$ and *not* $L \in \Delta_i$ then $F_{i+1} = F_i'$ and $\Delta_{i+1} = \Delta_i$.

C4. $L \in A$ and $L \notin \Delta_i$ and *not* $L \notin \Delta_i$, then
if there is a successful abductive derivation of *not* L,
then $F_{i+1} = F_i^-$ and $\Delta_{i+1} = \Delta'$,
otherwise $F_{i+1} = \{\leftarrow \phi_i'\} \cup F_i'$ and $\Delta_{i+1} = \Delta_i$.

$(\Delta_{i+1} = \Delta')$

$(not\ L, \Delta_i)$

$\vdots$

$(\varnothing, \Delta')$

Abductive phase

Example of an abductive proof

KB	A	G
p(X)← *not* q(X) q(X)← b(X)	b(a)	p(a)

Example of an abductive proof

KB

$p(X) \leftarrow \textit{not}\ q(X)$
$q(X) \leftarrow b(X)$
$\leftarrow \textit{not}\ p(X),\ p(X)$
$\leftarrow \textit{not}\ q(X),\ q(X)$
$\leftarrow \textit{not}\ b(X),\ b(X)$

A

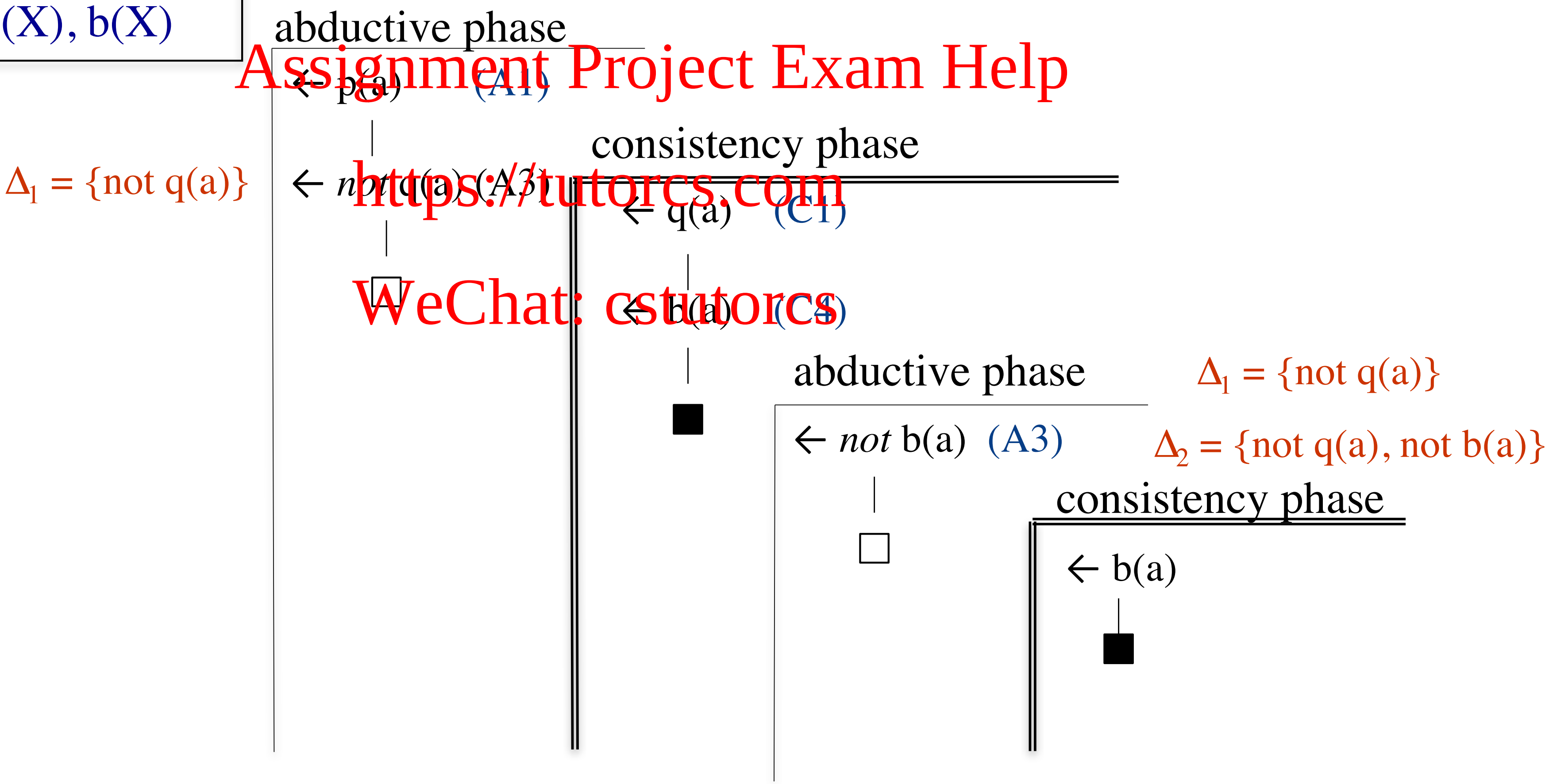
$b(a)$, $\textit{not}\ b(a)$
$\textit{not}\ p(a)$
$\textit{not}\ q(a)$

G

$p(a)$

Abductive solution

$\Delta = \{\text{not } q(a), \text{not } b(a)\}$



Semantic properties

- Knowledge assimilation through abduction
  - Addition of new information to KB:
    explanation of new information computed abductively, and adding to the KB

Two possible explanations

$$\text{KB} \left\{ \begin{array}{l} p \leftarrow q \\ p \\ r \leftarrow q \\ r \leftarrow s \end{array} \right. \quad \Longrightarrow \quad \begin{array}{l} \Delta_1 = \{q\} \\ \\ \Delta_2 = \{s\} \end{array}$$

r (new information)

Sometime q is preferred as it allows the inference of more information

Semantic properties

- Abduction is non-monotonic
  - default reasoning as abduction :
    new information can invalidate previous conclusions, when these are based on unproven assumption that are contradicted by the new information.

Making assumptions about applicability of a default rule is a form of abduction.

$\mathcal{F}$:
- bird(X) ← penguin(X)
- ¬ fly(X) ← penguin(X)
- penguin(tweety)
- bird(john)

$\mathcal{D}$: fly(X) ← bird(X)

KB:
- bird(X) ← penguin(X)
- not_fly(X) ← penguin(X)
- penguin(tweety)
- bird(john)
- fly(X) ← bird(X), birdsFly(X)

IC: ← birdsFly(X), not_fly(X)

Δ: birdsFly(john)

Semantic properties

- Abduction is non-monotonic
  - *abductive interpretation of NAF shows even further the suitability of abduction for default reasoning.*

Default assumptions expressed as abductive hypothesis on *not abnormality*.

$\mathcal{F}$
- bird(X) ← penguin(X)
- ¬ fly(X) ← penguin(X)
- penguin(tweety)
- bird(john)

$\mathcal{D}$
- fly(X) ← bird(X)

KB
- bird(X) ← penguin(X)
- abnormal(X) ← penguin(X)
- penguin(tweety)
- bird(john)
- fly(X) ← bird(X), not abnormal(X)

IC
- $\varnothing$

Δ
- not abnormal(john)

Semantic properties

- Similarity between abduction and NAF
  - NAF as abduction: *negative literals can be seen as abducibles, and can be assumed to be true provided that, together with the program, do not violate integrity constraints.*

Integrity constraints play an important role in capturing semantics of NAF.

$$KB \vdash Q \quad \text{iff} \quad Q \text{ has abductive solution in } \langle KB^*, A^*, I^* \rangle$$

$$KB \begin{cases} p(X) \leftarrow \text{not } q(X), \\ q(X) \leftarrow b(X) \end{cases} \Rightarrow \quad KB^* \begin{cases} p(X) \leftarrow q^*(X) \\ q(X) \leftarrow b(X) \end{cases} \quad IC^* \begin{cases} \leftarrow q^*(X), q(X) \\ \leftarrow p^*(X), p(X) \\ \leftarrow b^*(X), b(X) \end{cases}$$

$$A^* \; \{ q^*(X), p^*(X), b^*(X), b(X) \}$$

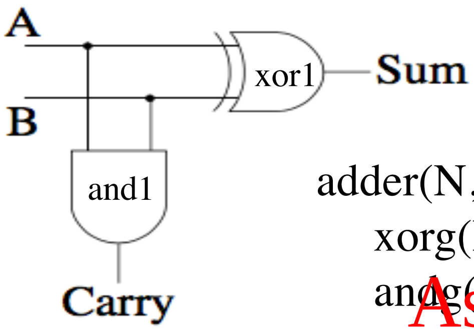
$$G = p(a)$$

$$\Delta = \{q^*(a), b^*(a)\}$$

Applications of Abduction

- Diagnosis problems
  - *Medical diagnosis:* background knowledge is the doctor's expertise, goals to explain are patient's symptoms, abducibles are all possible medical conditions, and abductive solutions are assumptions on specific medical conditions that explain patient's symptoms.
  - *Fault diagnosis:* find explanations for system's wrong behaviors. Goals are faulty traces of the system, knowledge base is description of the system behavior, integrity constraints are relevant constraint that the system has to maintain, abducibles are system's events.

Fault Diagnosis: example




```
adder(N, A, B, Sum, Carry):-
    xorg(N-xor1, A, B, Sum),
    andg(N-and1, A, B, Carry).
```

```
xor(0,1,1).
xor(1,0,1).
xor(1,1,0).
xor(0,0,0).
and(1,0,0).
and(0,1,0).
and(0,0,0).
and(1,1,1).
```

```
xorg(N, X,Y,Z):- xor(X,Y,Z).
xorg(N,0,0,1) :- fault(N, s1).
xorg(N,0,1,0) :- fault(N, s0).
xorg(N,1,0,0) :- fault(N, s0).
xorg(N,1,1,1) :- fault(N, s1).
andg(N, X,Y, Z):- and(X,Y,Z).
andg(N, 0,0, 1):- fault(N, s1).
andg(N, 1,0, 1):- fault(N, s1).
andg(N, 0,1, 1):- fault(N, s1).
andg(N, 1,1, 0):- fault(N, s0).
```


A = {fault(N, s0), fault(N, s1)}

G_1= adder(half_add,0,0,1,0) Δ_1 = {[fault(half_add, s1)]}

G_2= adder(half_add,0,1,0,1) Δ_2 = { [fault(half_add, s1), fault(half_add, s0)]) }

Abduction for planning

$$KB \begin{cases} have(X) \leftarrow buy(X) \\ have(X) \leftarrow hire(X) \\ have(X) \leftarrow borrow(X) \end{cases}$$

$$A = \{buy(\_), hire(\_), borrow(\_)\}$$

$$IC \begin{cases} \leftarrow hire(X), no\_have\_money \\ \leftarrow hire(car), not\ own(driving\_licence) \end{cases}$$

$$G \ \{ \ have(car)$$

$$\Delta_1 = \{hire(car), own(driving\_licence)\} \qquad \text{(plan1)}$$

$$\Delta_2 = \{borrow(car)\} \qquad \text{(plan2)}$$

$$\Delta_2 = \{buy(car)\} \qquad \text{(plan3)}$$

Summary

- Introduced the notion of abductive reasoning
- Desirable properties
  - Consistency of assumptions
  - Minimality of Explanation
- Algorithm
- Semantic Properties
  - Default reasoning
  - Abductive interpretation of NAF
- Some applications of abduction