Assumption-based reasoning = Abduction

- Abduction for artificial intelligence
- Abduction in logic: Theorist
- Abductive logic programming
- CIFF (lab)

Some references

Theorist:

- *A logical framework for default reasoning,* Poole
  In: AI Journal 36(1), 1988
- www.cs.ubc.ca/spider/poole/theorist.html

Abductive logic programming:

- *The role of abduction in logic programming* , Kakas&Kowalski&Toni
  In: Handbook of Logic in AI and LP, OUP, 1998 (available from my web page)
- *Abduction in logic programming*, Denecker&Kakas,
  In: Computation Logic: Logic Programming and Beyond – Essays in Honour of Robert Kowalski, LNAI 2407, Springer Verlag, 2002

CIFF:

- www.doc.ic.ac.uk/~ue/ciff/

Abduction: example

shoes are wet

shoes are wet if grass is wet
grass is wet if it rained
grass is wet if sprinkler was on

it rained

but what if: *cloudless sky*
and: *if cloudless sky & it rained then false* ?

Abduction is non-deterministic, fallacious, non-monotonic

Terminology

From B (e.g. shoes are wet)

B if A (e.g. *shoes are wet if it rained*)

infer A (e.g. it rained)

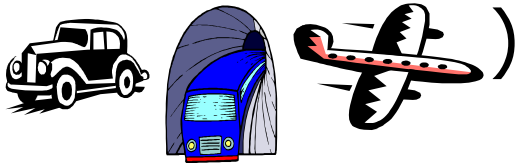
B is an ***observation***

A is an ***assumption/hypothesis/abducible*** that ***explains*** B

A is an ***explanation*** for B

Abduction for AI: applications (1)

Planning:

- observations are goals (e.g.)
- explanations are plans (e.g. )

Diagnosis:

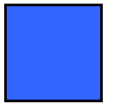
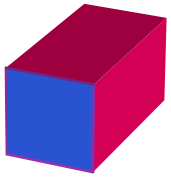
- observations are symptoms (e.g. toothache)
- explanations are diseases/faults (e.g. cavity)

Design:

- observations are design goals (e.g. pc)
- explanations are designs
  (e.g. processor + operating system)

Abduction for AI: applications (2)

Vision:

- observations are partial descriptions (e.g. )
- explanations are objects (e.g. or )

Natural language understanding:

- observations are ambiguous sentences

  (e.g. "The Milan office called…")
- explanations are interpretations

  (e.g. "John working for the office located in …road in Milan called…")

Abduction for AI: applications (3)

Knowledge assimilation:

- observations are new data

  (e.g. Ann is the sibling of Bob)
- explanations are data deriving them

  (e.g. Ann and Bob share a parent)

Default reasoning:

- observations are predictions

  (e.g. Tweety flies / Tweety does not fly)
- explanations are default rules

  (e.g. birds fly, and we know that Tweety is a bird /
  penguins do not fly, and we know that Tweety is a penguin)

Logic (background)

- Terms
- Atoms
- Literals
- Clauses
- Horn clauses
- Sentences (propositional/first order logic)
- Entailment $\models$ (propositional/first order logic)

Abduction in logic: Theorist

Given **T** (*theory presentation*), ***FOL theory***

H (*candidate hypotheses*), set of ground ***FOL sentences***

O (*observation*), ***FOL sentence***

E (*explanation*) is such that

1) $T \cup E$ **entails** O

2) $T \cup E$ is **consistent**

(equivalently $T \cup E$ **does not entail false**)

3) $E \subseteq H$

Note: "entailment" is FOL entailment

Without loss of generality, H is a set of (ground) *atoms*

Indeed, for every non-atomic hypothesis s in H
$s \leftarrow h$ can be added to T, giving T'
with h a new hypothesis in H'

Then, given O,
explanations of O wrt T and H are in
one-to-one correspondence with
explanations of O wrt T' and H'

Theorist: example of diagnosis

T: wobbly-wheel ***if*** broken-spokes ***or*** flat-tyre
flat-tyre ***if*** leaky-valve ***or*** punctured-tube
not leaky-valve
H: flat-tyre, broken-spokes, leaky-valve, punctured-tube
O: wobbly-wheel

E: {broken-spokes}, {punctured-tube} ✔
{wobbly-wheel}, {leaky-valve} ✗
{leaky-valve, broken-spokes} ✗
{broken-spokes, punctured-tube} ✔
{flat-tyre} ✔

Additional requirements on E

In addition, **E** might be required to be

basic, e.g. **O**=p, **T**={p ***if*** q, q ***if*** r, p ***if*** s}

E={r} is *basic*

E={q} and **E**={p} are *not*

minimal, e.g. **E**={r} is *minimal*

E={r,q} is *not*

optimal, e.g. utility(r)=10, utility(s)=100

E={s} is *optimal*

E={r} is *not*

Basic and minimal explanations

- E is **basic** for O iff there exists no non-*trivial* * explanation E' for O such that E' is an explanation for E

* E is a *trivial* explanation of O iff E entails O

  e.g (any superset of) any E is a trivial explanation of itself

  Note: H can be chosen so that every E is basic

- E is **minimal** for O iff there exists no E' such that E' $\subset$ E and E' is an explanation for O

  Note: some computational techniques guarantee "local" minimality of explanation

Exercises (Theorist)

1. Which explanations are basic/minimal in the example on slide 9?
2. Example of knowledge assimilation:
   Give all possible explanations for **O:** sibling(ann,bob), given

T: sibling(X,Y) ***if*** parent(Z,X) ***and*** parent(Z,Y)
parent(X,Y) ***if*** mother(X,Y)
parent(X,Y) ***if*** father(X,Y)
mother(mary,ann)
father(john,ann)

H={mother(t,s), father(t,s) | t,s ground terms}

Abductive logic programming

- SLD resolution *(background)*
- Abductive SLD: ASLD
- Abductive SLD with integrity constraint checking: ASLDIC
- Abductive SLD with integrity constraint checking and negation as failure: ASLDICN *(only mentioned)*
- CIFF: an abductive system

Logic programs

A (normal) logic program is a set of

- (implicitly universally quantified) rules of the form

  A **if** L_1 **&** ... **&** L_n, with A atom and L_i literals, n >0
  [namely, either L_i = B or L_i = **not** B, with B atom]

  A is called the *head* and

  L_1 **&** ... **&** L_n is called the *body* of the rule

- (implicitly universally quantified) facts A, with A atom

A definite logic program is a logic program without negation
(set of Horn clauses)

The negation **not** is to be interpreted via the
principle of negation as failure : **not p** holds if **p** fails to hold
(different from classical negation, see the program p **if not** q)

Abductive logic programming - version 1

T is a ***definite*** **logic program**

H is a set of ***undefined*** ground **atoms**

(without loss of generality)

O is a conjunction of (existentially quantified) **atoms**

E is such that

1) T ∪ E "entails" O

2) T ∪ E does not "entail" false

3) E ⊆ H

If some atom h in H is defined in T:

Let h **if** B be a clause in **T**

Let h′ be a new atom, not occuring in **T**, and let

T′ be **T**∪{h **if** h′} and **H′** be **H**-{h}∪{h′}.

Then **<T′, H′>** is equivalent to **<T,H>**,

in that for every **O**,

there exists an explanation **E** for **O** wrt **<T,H>** iff

there exist an explanation **E′** for **O** wrt **<T′, H′>**.

Semantics of definite logic programs

- The meaning (semantics) of a definite logic program is given by its *least Herbrand model* (LHM in short).

  e.g. for P_{simple} : p **if** q
  s **if** q **and** r
  q

  the LHM is {q, p}

- Given a definite logic program P, LHM(P)= least fixed point of $\mathcal{T}_P = \mathcal{T}_P \uparrow \omega$, where the operator $\mathcal{T}_P$ is defined by:

  $\mathcal{T}_P$ (I)= {*A* | *A* is an atom, *A* ***if*** *B* $\in$ ground(P) and B $\subseteq$ I}

  e.g. for $P=P_{simple}$

  $\mathcal{T}_P$ ({})={q}; $\mathcal{T}_P$ ({q})={q,p}; $\mathcal{T}_P$ ({q,p})={q,p}: LHM

Abductive logic programming-version1 revisited

"Entailment" is truth in the LHM,

namely if M is the LHM of T $\cup$ E, then:

1) becomes *O is true in M*
2) becomes *false is not true in M*,
   which is always the case

Exercise (ALP - version 1)

Given the abductive logic program <T,H> with

T: p(X) **if** q(X,Y) **and** s(Y)
s(X) **if** a(X) **and** b(X,Y)
s(X) **if** c(X)
q(3,4)
q(X,Y) **if** q(Y,X)
t(X) **if** s(X)

H: all ground instances of the predicates a, b, c

list all possible explanations for the observations,

O_1: p(3) O_2: p(5) O_3: p(4) **and** t(6)

and show why they are indeed explanations.

Which explanations are minimal? Which are basic?

Computation of explanations for observations?

Abductive SLD (ASLD in short):

Extension of SLD-resolution

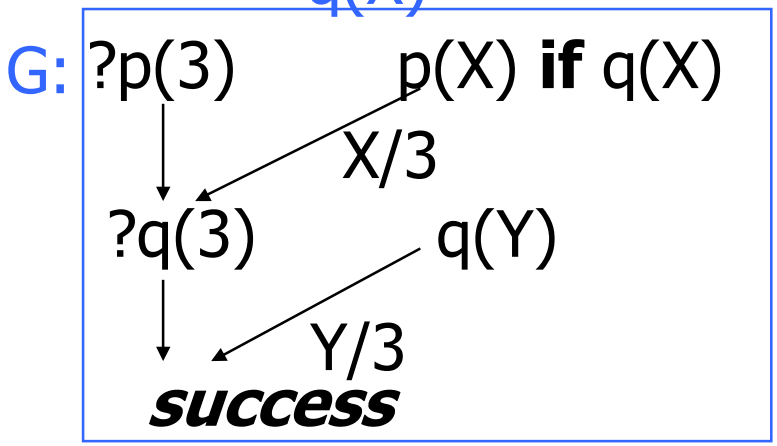
(*Linear* resolution with *Selection* function for *Definite* logic programs)

SLD-Resolution(1)

- Given a program P, an *SLD-derivation* for a goal/observation G (wrt a *selection rule* R) consists of
  - a sequence of goals $<G_0=G, G_1, \ldots>$,
  - a sequence of mgus $<\theta_1, \theta_2, \ldots>$ and
  - a sequence of (variants of) rules in P $<C_1, C_2, \ldots>$

  such that G_{i+1} is the resolvent of G_i and C_{i+1} via θ_{i+1} on the atom selected by R in G_i and the head of C_{i+1}

e.g. P : p(X) **if** q(X)
q(X)



$<G_0=p(3), G_1=q(3), G_2=empty>$
$<\theta_1 =\{X=3\}, \theta_2 =\{Y=3\}>$
$<C_1 =p(X)$ if $q(X), C_2 =q(Y)>$
R is any (e.g. left-most)

SLD-resolution (2)

success of an SLD-derivation

amounts to being left with proving the

empty goal (observation)

and is often represented simply by □

SLD-Resolution (3)

- A (*successful*) *SLD-refutation* for a goal G (wrt R) is a finite SLD-derivation (wrt R) $<G_0, G_1, \ldots, G_n>, <\theta_1, \theta_2, \ldots, \theta_n>, <C_1, C_2, \ldots, C_n>$ with $G_0 = G$ & $G_n = \square$.

- The refutation *computes the answer* θ obtained by restricting the composition of $\theta_1, \theta_2, \ldots, \theta_n$ to the variables of G (*thus we mostly omit all other variable substitutions*).

- We represent a refutation in short as $<(G,\theta_0), (G_1\ \theta_1), \ldots, (G_{n-1}, \theta_{n-1}), (\square, \theta_n)>$, with $\theta_0 = \{\}$

e.g. P : p(X) **if** q(X)
q(4)

G: ?p(Z)

? p(Z)
↓
? q(Z)
↓ Z/4
□ with Z=4

Short graphical representation

SLD-Resolution(4)

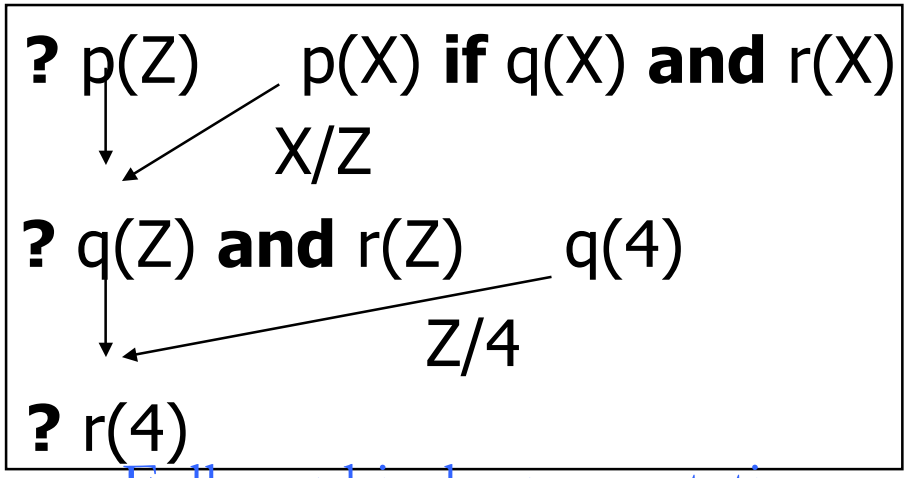
- Given a program P and a goal G, if there exists an SLD-refutation for G (wrt some selection rule R) computing the answer θ then Gθ is true in the LHM of P (soundness of SLD-resolution)

- Given a program P and a goal G, if there exists a substitution σ such that Gσ is true in the LHM of P then there exists a selection rule R and an SLD-refutation for G wrt R computing some answer θ such that $\sigma = \theta\theta'$, for some substitution θ' (completeness of SLD-resolution)

SLD-Resolution(5)

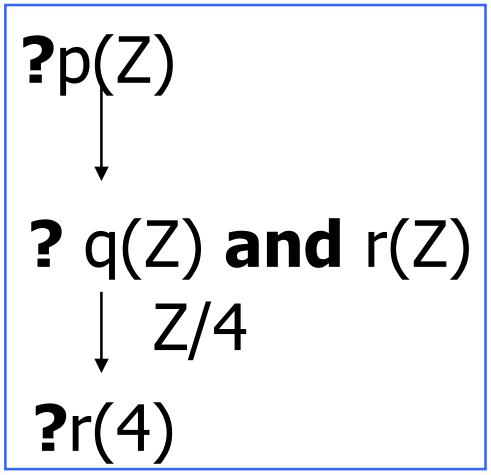
- A *finitely failed SLD-derivation* for a goal G (wrt a selection rule R) is a finite SLD-derivation (wrt R) $<G_0, G_1, \ldots, G_n>$, $<\theta_1, \theta_2, \ldots, \theta_n>$ and $<C_1, C_2, \ldots, C_n>$ with $G_0 = G$ and $G_n \neq \square$ such that there exist no resolvent of G_n.

e.g. P : p(X) **if** q(X) **and** r(X)
q(4)



Full graphical representation

Short graphical representation



SLD-resolution (6)

failure of an SLD-derivation

amounts to being left with proving a

non-empty goal (observation)

which cannot be proven

and is often represented simply by ✗

E.g. (example from previous slide)

?p(Z)

? q(Z) **and** r(Z)

?r(4)

✗

SLD-Resolution(7)

- Given a program P, the *SLD-tree* for a goal G (wrt a selection rule R) is such that:
  - every node is a goal (possibly empty)
  - the root is G
  - let **?**A_1 **and**…**and** A_m be a (non-empty) node and A_i be selected by R; then the node has a child for every rule in P whose head unifies with A_i; the child is the resolvent
  - empty nodes have no children

e.g. Build the SLD-tree for P : p(X) **if** q(X)
p(X) **if** r(X)
q(4)

G: **?**p(Y)

ASLD-resolution (1)

- Given an abductive logic program <T,H>, an *abductive* SLD-derivation for an observation O (wrt a *safe* selection rule R) consists of
  - a sequence of goals $<G_0=O, G_1, \ldots>$,
  - a sequence of mgus $< \theta_0=\{\}, \theta_1, \theta_2, \ldots>$,
  - a sequence of (variants of) rules in T $< C_0=\{\}, C_1, C_2, \ldots>$
  - *a sequence of sets of hypotheses* $<E_0=\{\}, E_1, E_2, \ldots>$

  such that, if A is selected by R in G_i, then

1. If $A \in H$ and $A \in E_i$,
   then $G_{i+1}=G_i-\{A\}$, $E_{i+1}=E_i$, $\theta_{i+1}=\{\}$, $C_{i+1}=A$,
2. If $A \in H$ but $A \notin E_i$,
   then $G_{i+1}=G_i-\{A\}$, $E_{i+1}=E_i \cup \{A\}$, $\theta_{i+1}=\{\}$, $C_{i+1}=A$,
3. If $A \notin H$, then $E_{i+1}=E_i$ and
   G_{i+1} is the resolvent of G_i and C_{i+1} via θ_{i+1} on the atom selected by R in G_i and the head of C_{i+1}

ASLD-resolution (2)

- A safe selection rule selects any atom A of an *abducible predicate* only if A is ground, where a predicate is abducible iff it is the predicate of an abducible atom in H
- *Abductive* SLD-refutation/successful derivation/finitely failed derivation/tree.
- We represent an abductive refutation in short as $<(O,\theta_0,E_0), \ldots ,(G_{n-1}, \theta_{n-1},E_{n-1}), (\square, \theta_n,E_n)>$ with $\theta_0=\{\}$, $E_0=\{\}$
- We say that the refutation:
  - computes the answer θ obtained by restricting the composition of $\theta_1,\ldots,\theta_n$ to the variables of O
  - *computes the set of hypotheses* E_n for O
- We abbreviate "Abductive SLD" to "ASLD"

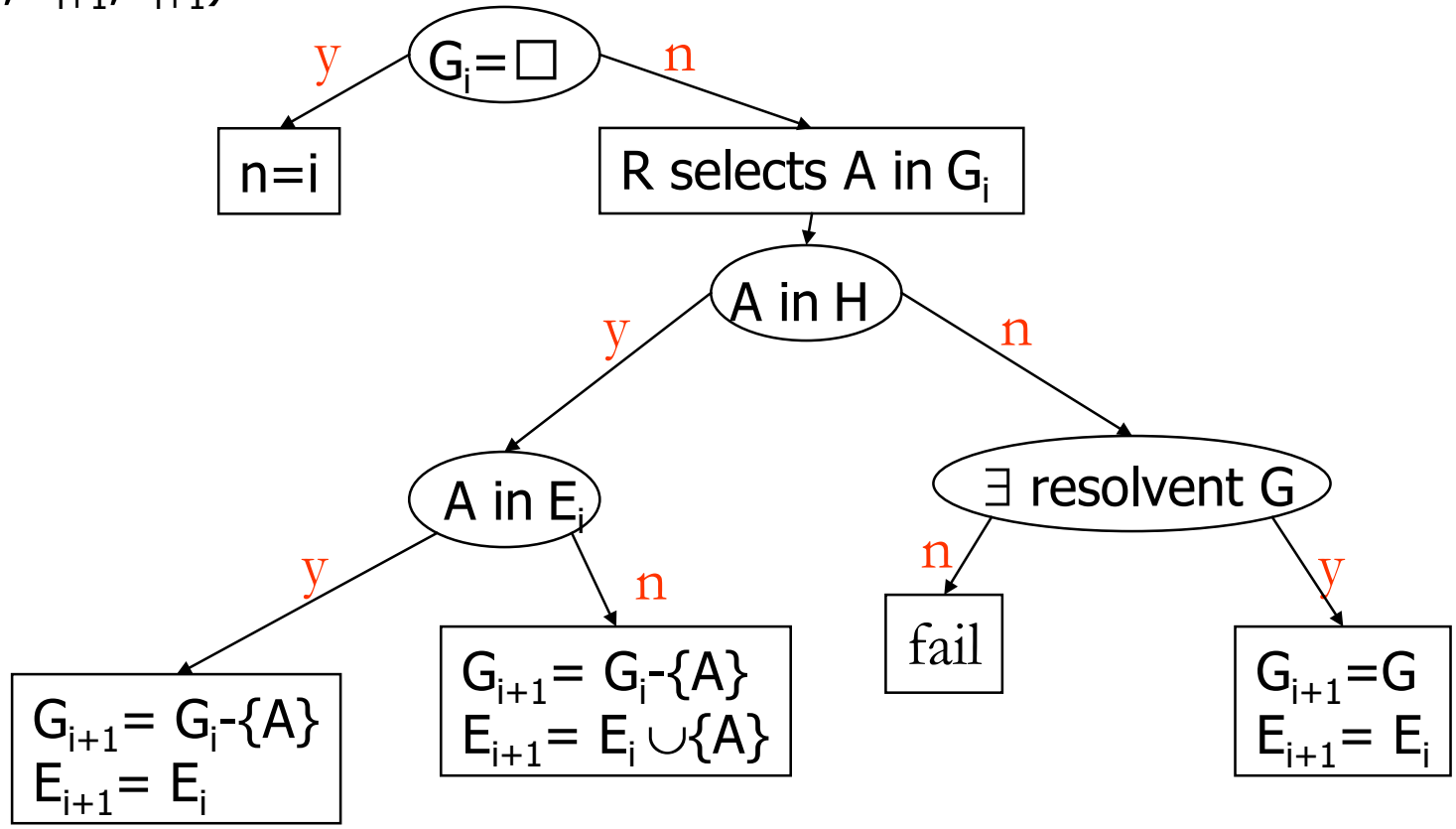
ASLD at a glance

We want to build a refutation

$<(O,\theta_0,E_0), \ldots ,(G_{n-1}, \theta_{n-1},E_{n-1}), (\square, \theta_n,E_n)>$ with $\theta_0=\{\}$, $E_0=\{\}$

Suppose we have built a derivation till (G_i, θ_i,E_i), $0 \leq i < n$.

Then $(G_{i+1}, \theta_{i+1},E_{i+1})$ is constructed as follows:



Example (ASLD for diagnosis)

T: wobbly-wheel(X) ***if*** broken-spokes(X)
wobbly-wheel(X) ***if*** flat-tyre(X)
flat-tyre(X) ***if*** leaky-valve(X)
flat-tyre(X) ***if*** punctured-tube(X)
H: broken-spokes(b), leaky-valve(b), punctured-tube(b)
O: wobbly-wheel(b)

<(?ww(b),{},{}), (?bs(b),{},{}),(□,{},{bs(b)})>
<(?ww(b),{},{}), (?ft(b),{},{}), (?lv(b),{},{}),(□,{},{lv(b)})>
<(?ww(b),{},{}), (?ft(b),{},{}), (?pt(b),{},{}),(□,{},{pt(b)})>

?ww(b)	?ww(b)	?ww(b)
?bs(b)	?ft(b)	?ft(b)
□	?lv(b)	?pt{b)
{bs(b)}	□	□
	{lv(b)}	{pt(b)}

Exercise: draw the ASLD-tree for ?ww(b)

Soundness and (weak) completeness of abductive SLD-resolution

Given abductive logic program <T,H> and observation O:

- if there exists an ASLD-refutation for O (wrt some selection rule R) computing the answer θ and the set of hypotheses E, then E is an explanation for Oθ (soundness of ASLD-resolution)

- if T is such that for every clause C in T, *every variable occuring in the body of C also occurs in the head of C*, then: if *O is ground* and there exists a set of hypotheses E such that E is an explanation of O, then there exists a (safe) selection rule R and an ASLD-refutation for O wrt R computing some set of hypotheses E' such that $E' \subseteq E$ (weak completeness of ASLD-resolution)

Example
(weak completeness of ASLD)

T: p(X) **if** a(X)
   r(X) **if** b(X)
H: all ground instances of the predicates a and b
O: p(3)

For E={a(3),b(4)}, E is an explanation for O,
but there exists no ASLD-refutation computing E.
However, there exists an ASLD-refutation computing E'={a(3)}.

Strong completeness of abductive SLD-resolution does not hold

Given abductive logic program <T,H> and observation O:

- if there exists a substitution σ and a set of hypotheses E such that E is an explanation of Oσ, then there exists a selection rule R and an ASLD-refutation for O wrt R computing some answer θ and some set of hypotheses E' such that $\sigma = \theta\theta'$, for some substitution θ', and $E' \subseteq E$ (strong completeness of ASLD-resolution)

Counterexample:
T: p(X) if a(Y)
H: all ground instances of the predicate a
O: p(3)

Minimality of explanations computed by abductive SLD-resolution

Given abductive logic program <T,H> and observation O:

- if there exists an abductive SLD-refutation for O (wrt some selection rule R) computing the answer θ and the set of hypotheses E, then E is a locally minimal explanation for Oθ, i.e. E is minimal wrt <T',H>, with T'$\subseteq$T

e.g. T: p **if** a H: a,b O: p

p **if** a **and** b

E_1={a} and E_2={a,b} both computed by abductive SLD

E_1 is minimal

E_2 is not, but it is minimal wrt T': p **if** a **and** b

"Basicality" of explanations computed by abductive SLD-resolution

Given abductive logic program <T,H> and observation O:

- If, there exists an abductive SLD-refutation for O (wrt some selection rule R) computing the answer θ and the set of hypotheses E, then E is a basic explanation for Oθ.

[This is due to hypotheses being undefined in T]

e.g. T: p **if** a H: a O: p

E={a} computed by abductive SLD is basic

Exercise (ASLD)

1. Given the following abductive logic programs <T,H> and observations O, give all sets of hypotheses computed by ASLD

   i. T: p(X) **if** q(X,3)

      q(X,Y) **if** a(Y)

      H: all ground instances of the predicate a

      O: p(Z)

   ii. T: p(X) **if** a(X)

      H: all ground instances of the predicate a

      O: p(Z)

   Which explanations are minimal? Which are basic?

Abduction with integrity constraints

E might be required to satisfy some given *domain-specific properties* (**integrity constraints** in DBs):

e.g. **O:** sibling(ann,bob)

T: sibling(X,Y) ***if*** parent(Z,X) ***and*** parent(Z,Y)

parent(X,Y) ***if*** mother(X,Y)

parent(X,Y) ***if*** father(X,Y)

mother(mary,ann)

father(john,ann)

H={mother(t,s), father(t,s) | t,s ground terms}

E={mother(john,bob)} violates the property that

if mother(X,Y) ***and*** father(X,Z) ***then false***

Integrity constraints satisfaction

A given knowledge base KB (e.g. T $\cup$ E)
satisfies a set of integrity constaints (sentences) IC if:
consistency view: KB$\cup$IC is "consistent"
theoremhood view: KB "entails" IC
epistemic/ meta-level view: IC is "known/believed" in KB

e.g. IC: ***if*** employee(X) ***then*** $\exists$ Y insurance-no(X,Y)
$\Rightarrow$KB={employee(mary)} satisfies IC under *consistency view,*
but not under *theoremhood* and *epistemic views*
$\Rightarrow$KB={} satisfies IC under *consistency* and *epistemic views,*
but not under *theoremhood view*

Abductive logic programming – version 2

Given:

T (theory presentation) is a definite logic program

H (candidate hypotheses) is a set of undefined ground atoms

IC (integrity constraints) is a set of sentences

O (observation) is a conjunction of atoms

Then: **E** (explanation) is such that

1) $T \cup E$ entails O

2') $T \cup E$ "satisfies" IC

3) $E \subseteq H$

- Let M be the least Herbrand model of $T \cup E$; then:
  - 1) becomes O is true in M;
  - 2') becomes IC are true in M (*epistemic view*)
- We will assume that IC is a set of ***denials*** of the form
  if A_1 **and** … **and** A_m **then false**, A_i atom, m>0

Example (ALP-version 2 for diagnosis)

T: wobbly-wheel ***if*** broken-spokes
wobbly-wheel ***if*** flat-tyre
flat-tyre ***if*** leaky-valve
flat-tyre ***if*** punctured-tube

H: broken-spokes, leaky-valve, punctured-tube

IC: if *leaky-valve* ***then false***

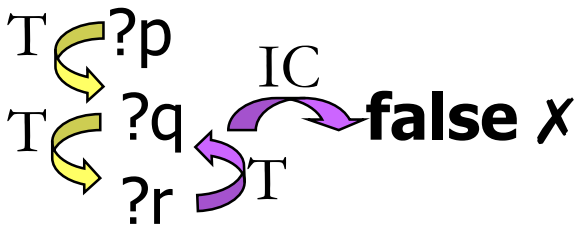
O: wobbly-wheel

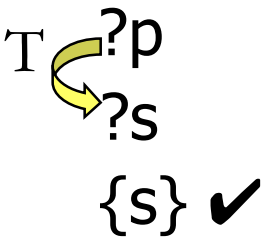
E: {broken-spokes}, {punctured-tube} ✔
{broken-spokes, punctured-tube} ✔
{leaky-valve} ✗
{leaky-valve, broken-spokes} ✗

Computation of explanations in the presence of integrity constraints

Given T,H,IC and O, an explanation
E (for O) can be computed via interleaving:
backward reasoning (e.g. via ASLD) with T to generate E
forward reasoning with T and IC to test E

e.g. T={p **if** q, q **if** r, p **if** s}
H={r,s}
IC={**if** q **then false**}
O=p





ASLD with testing of IC (ASLDIC)

Assume IC is a set of denials, of the form

if A_1 **and** … **and** A_m **then false**, A_i atom, m>0

with at least one A_i an atom of an *abducible predicate**

- *abductive* (ASLD) derivations/refutations to *prove* observations/sentences (condition 1))

  $<(O, \{\}), \ldots, (G_{n-1}, E_{n-1}), (\square, E_n)>$
- *consistency* derivations/refutations to *disprove* (residues of) integrity constraints (condition 2'))

  $<(S, E_o), \ldots, (S_{n-1}, E_{n-1}), (\{\}, E_n)>$

**Abducible predicates* are predicates of hypotheses in H

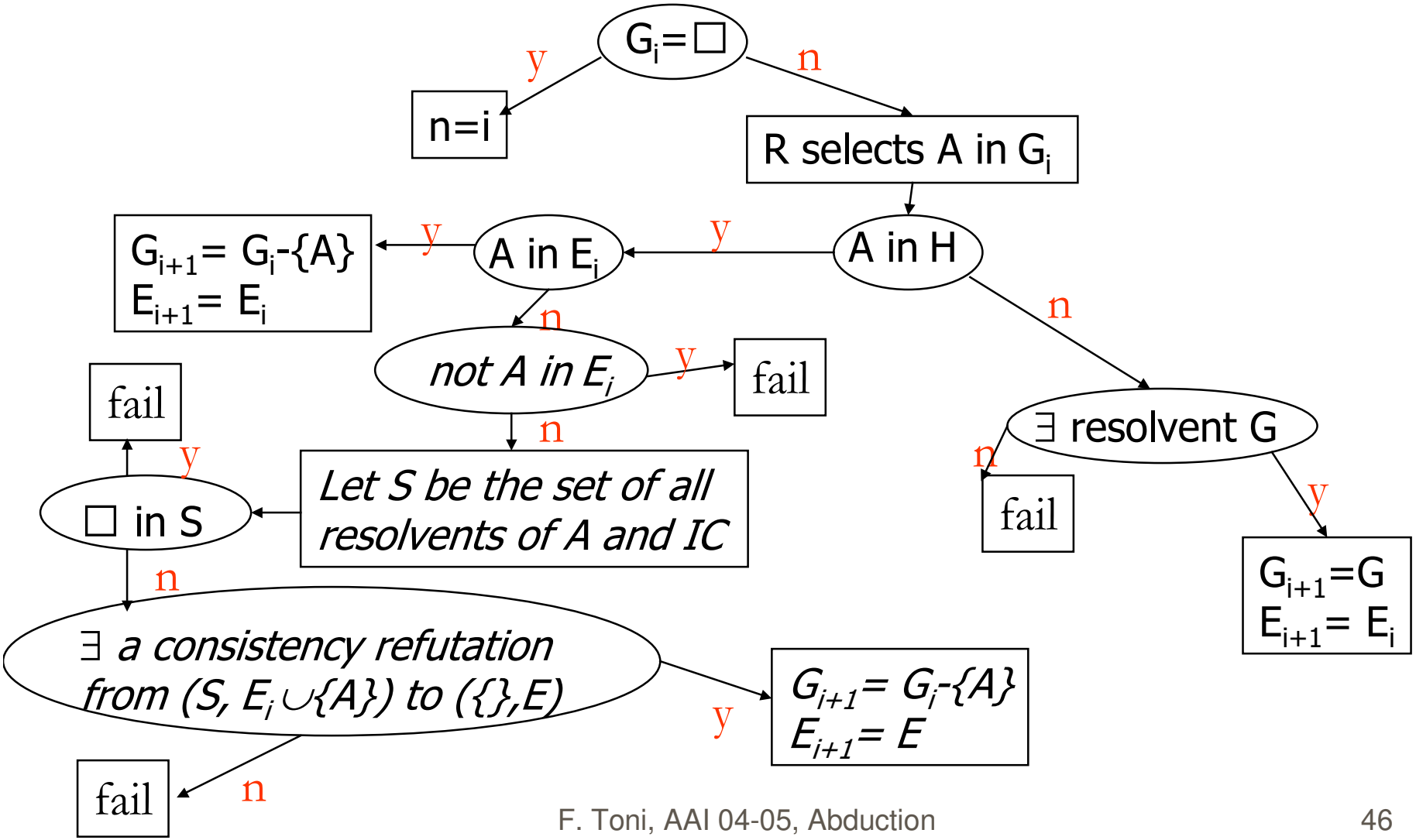
Note: we ignore mgus for simplicity

Note: E_i may contain negations of hypotheses

Abductive derivation/refutation at a glance

We want to construct $<(O, \{\}), \ldots, (G_{n-1}, E_{n-1}), (\square, E_n)>$

Given (G_i, E_i), $0 \leq i < n$, how do we construct (G_{i+1}, E_{i+1})?



Abductive derivation/refutation

- Given an abductive logic program <T,H,IC>, an abductive derivation/refutation for an observation O (wrt R) is

  $<(O, \{\}), \ldots, (G_n, E_n)> / <(O, \{\}), \ldots (\square, E_n)>$

  such that, if A is selected by R in G_i, $0 \leq i < n$, then

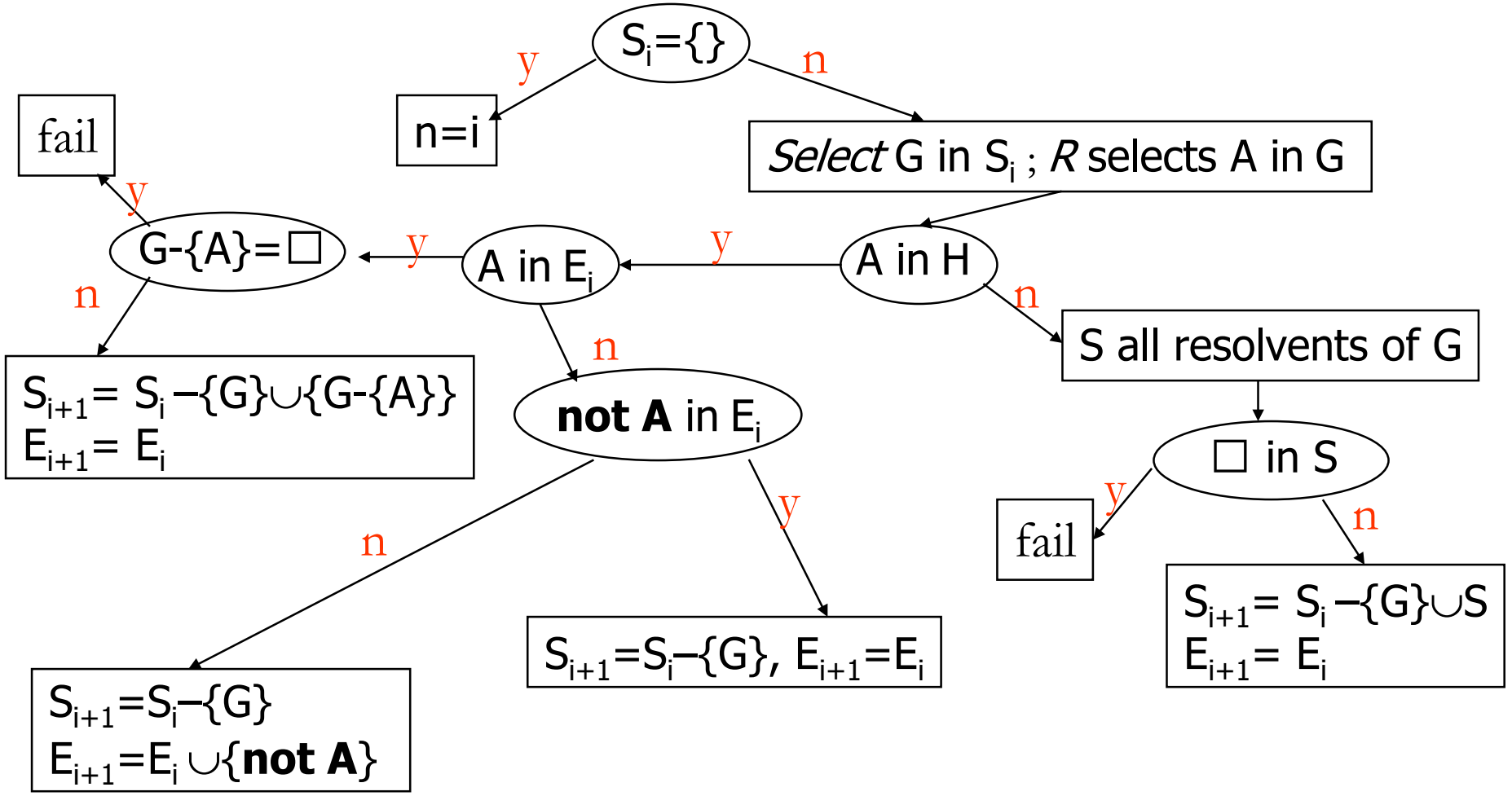
  1. If $A \in H$ and $A \in E_i$, then $G_{i+1}=G_i-\{A\}$, $E_{i+1}=E_i$,
  2. If $A \in H$ and $A \notin E_i$ and not $A \notin E_i$ let S be the set of all resolvents of A with IC; if $\square$ is not in S and there exists a *consistency refutation* $<(S, E_i \cup \{A\}), \ldots (\{\},E)>$ then $G_{i+1}=G_i-\{A\}$, $E_{i+1}=E$,
  3. If $A \notin H$, then $E_{i+1}=E_i$ and G_{i+1} is the resolvent of G_i and a rule/fact C in T on the atom selected by R in G_i and the head of C

- An abductive refutation $<(O, \{\}), \ldots (\square, E_n)>$ is said to compute the set of hypotheses $E_n \cap H$ (and an answer θ)

Consistency derivation/refutation at a glance

We want to construct $<(S_0, E_0), \ldots, (S_{n-1}, E_{n-1}), (\{\}, E_n)>$

Given (S_i, E_i), $0 \leq i < n$, how do we construct (S_{i+1}, E_{i+1})?



Consistency derivation/refutation

- Given an abductive logic program <T,H,IC>, a consistency derivation/refutation (wrt R) is

  $<<(S_0, E_0), \ldots, (S_n,E_n)>>$ / $<<(S_0, E_0),\ldots (\{\},E_n)>>$

  such that, if G is *selected* in S_i, $0 \leq i < n$, and A is selected by R in G, then

  1. If $A \in H$ and $A \in E_i$,

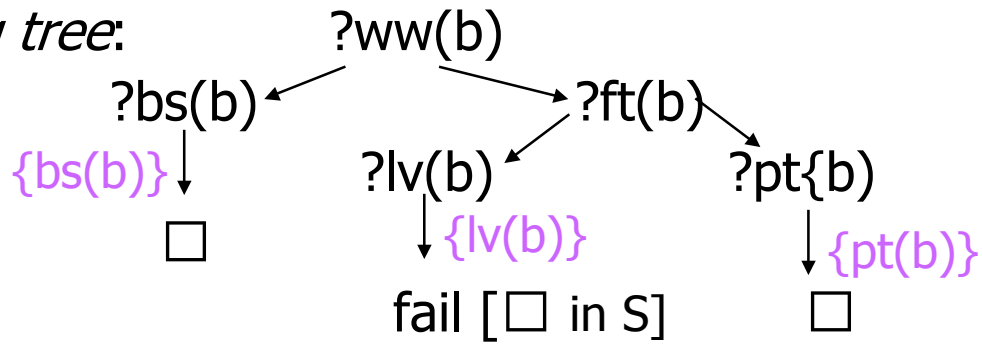
     then, if $G-\{A\} \neq \square$, $S_{i+1}=S_i - \{G\} \cup \{G-\{A\}\}$, $E_{i+1}=E_i$
  2. If $A \in H$, $A \notin E_i$ and **not A** $\in E_i$, then $S_{i+1}=S_i - \{G\}$, $E_{i+1}=E_i$
  3. If $A \in H$, $A \notin E_i$ and **not A** $\notin E_i$,

     then $S_{i+1}=S_i - \{G\}$, $E_{i+1}=E_i \cup \{$**not A**$\}$
  4. If $A \notin H$, then let S be the set of all resolvents of G and a rule/fact C in T on the atom A selected by R in G and the head of C. Then, if $\square \notin S$, $S_{i+1}=S_i - \{G\} \cup S$, $E_{i+1} =E_i$

Example (ASLDIC for diagnosis)

T: wobbly-wheel(X) ***if*** broken-spokes(X)
wobbly-wheel(X) ***if*** flat-tyre(X)
flat-tyre(X) ***if*** leaky-valve(X)
flat-tyre(X) ***if*** punctured-tube(X)
H: broken-spokes(b), leaky-valve(b), punctured-tube(b)
IC: **if** leaky-valve **then false**
O: wobbly-wheel(b)
<(?ww(b),{}), (?bs(b),{}),(□,{bs(b)})> refutation
<(?ww(b),{}), (?ft(b),{}), (?lv(b),{})> finitely failed
<(?ww(b),{}), (?ft(b),{}), (?pt(b),{}),(□,{pt(b)})> refutation

Corresponding tree:



Example (ASLDIC)

T: p(X) ***if*** q(X) **and** b(X)
q(X) ***if*** a(X)
q(X) ***if*** c(X)
IC: **if** a(X) **and** b(X) **then false**
H: all ground atoms of the predicates a,b,c
O: p(3)

<(?p(3),{}),(?q(3) and b(3),{}),(?a(3) and b(3),{}),(?b(3),{a(3),not b(3)})>

since <<({?b(3)},{a(3)}),({},{a(3),not b(3)})>>

<(?p(3),{}),(?q(3) and b(3),{}),(?c(3) and b(3),{}),
(?b(3),{c(3)}),(□,{c(3),b(3),not a(3)})>

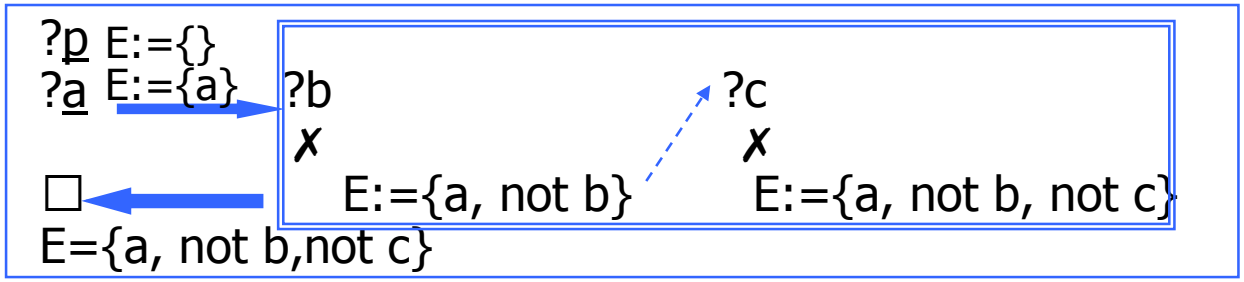
since <<({?a(3)},{c(3),b(3)}),({},{c(3),b(3),not a(3)})>>

Thus {c(3),b(3)} is the computed set of hypotheses

ASLDIC: Graphical conventions

T: p **if** a
H: a,b,c
IC: **if** a **and** b **then false**
if a **and** c **then false**
O: p

R: left-most selection rule



Stands for <(?p,{}),(?a,{}),(□,{a,not b, not c})>
<<({?b,?c},{a}),({?c},{a,not b}),({},{a,not b, not c})>>

Exercise: redo example on slide 51 wrt these graphical conventions

Soundness and (lack of) completeness of ASLDIC

Given abductive logic program <T,H,IC> and observation O:

- if there exists an abductive refutation for O (wrt some selection rule R) computing the answer θ and the set of hypotheses E, then E is an explanation for Oθ (soundness of ASLDIC-resolution)

- Counterexample to completeness of ASLDIC-resolution:
  T: q **if** a H:a IC: **if** a **and** p **then false** O: q
  p **if** p
  - E={a} (minimal) explanation for O
    [since LHM(T$\cup${a})={a,q}, and IC true in {a,q}]
  - E is not computed by ASLDIC

Exercise (ASLDIC)

Given the following abductive logic program <T,H,IC> and observation O, give all sets of hypotheses computed by ASLDIC

T: p(X) **if** q(X) **and** a(X)

q(X) **if** b(X)

q(X) **if** c(X)

r(X) **if** b(X)

H: all ground instances of the predicates a,b,c

IC: **if** a(X) **and** r(X) **then false**

O: p(3)

Abductive logic programming – version 3

Given: **T** (theory presentation) is a ***normal logic program***

H (candidate hypotheses) is a set of undefined ground atoms

IC (integrity constraints) is a set of sentences

O (observation) is a conjunction of **literals**

Then: **E** (explanation) is such that

1) T ∪ E "*entails*" O

2') T ∪ E "*satisfies*" IC

3) E ⊆H

- There are many possible notions of "*candidate*" model for a normal logic program. Let M be some "canonical" model of T ∪ E; then:
  - 1) becomes O is true in M;
  - 2') becomes IC are true in M (*epistemic view*)
- We will assume that IC is a set of denials of the form

  if L_1 **and** … **and** L_m **then false**, L_i **literal**, m>0, with at least one L_i an atom of an abducible predicate

Example (ALP version 3 for diagnosis)

Given **T:** wobbly-wheel **if** broken-spokes
wobbly-wheel **if** flat-tyre
flat-tyre **if** leaky-valve
flat-tyre **if** punctured-tube
flat-tyre **if not** air-in-tyre

H: broken-spokes, leaky-valve, punctured-tube, air-in-tyre

IC: **if** leaky-valve **then false**

and **O:** wobbly-wheel

Then E={broken-spokes} ✓ M(T∪E)={bs,ww}
E={punctured-tube} ✓ M(T∪E)={pt,ft,,ww}
E={leaky-valve} ✗ M(T∪E)={lv,ft,ww}
E={} ✓ (implicitly, **not** air-in-tyre) M(T∪E)={ft,ww}

where M is the *perfect model*

Example (ALP version 3 for simple planning)

Given **T:** have(X) **if** buy(X)

have(X) **if** steal(X)

H: buy, steal

IC: **if** **not** money **and** buy(X) **then** **false**

and **O:** have(PC)

then E={steal(PC)} ✓

E={buy(PC)} ✗

Example (ALP version 3 for planning)

Given **T:** holds(P,T) **if** holds-initially(P) **&** T≥0
& not broken(P,0,T)
holds(P,T) **if** happens(A,T') **&** T'<T
& initiates(A,T',P) **& not** broken(P,T',T)
broken(P,T',T) **if** happens(A,T")
& terminates(A,T",P) **&** T'<T"<T
holds-initially(in)
initiates(park,T,in)
terminates(steal,T,in)
initiates(steal,T,out)
preconditions(steal,in)
1>0
2>1 …

H: happens(e,t) for all ground terms e,t

IC: **if** happens(E,T) **&** preconditions(E,P) **then** holds(P,T)

Variant of (simplified) Event Calculus

ASLDIC with negation (ASLDICN)

No time to define it –

See the paper Kakas-Kowalski-Toni 1998:

ASLDICN is basically

the Kakas-Mancarella procedure

Example (ASLDICN)

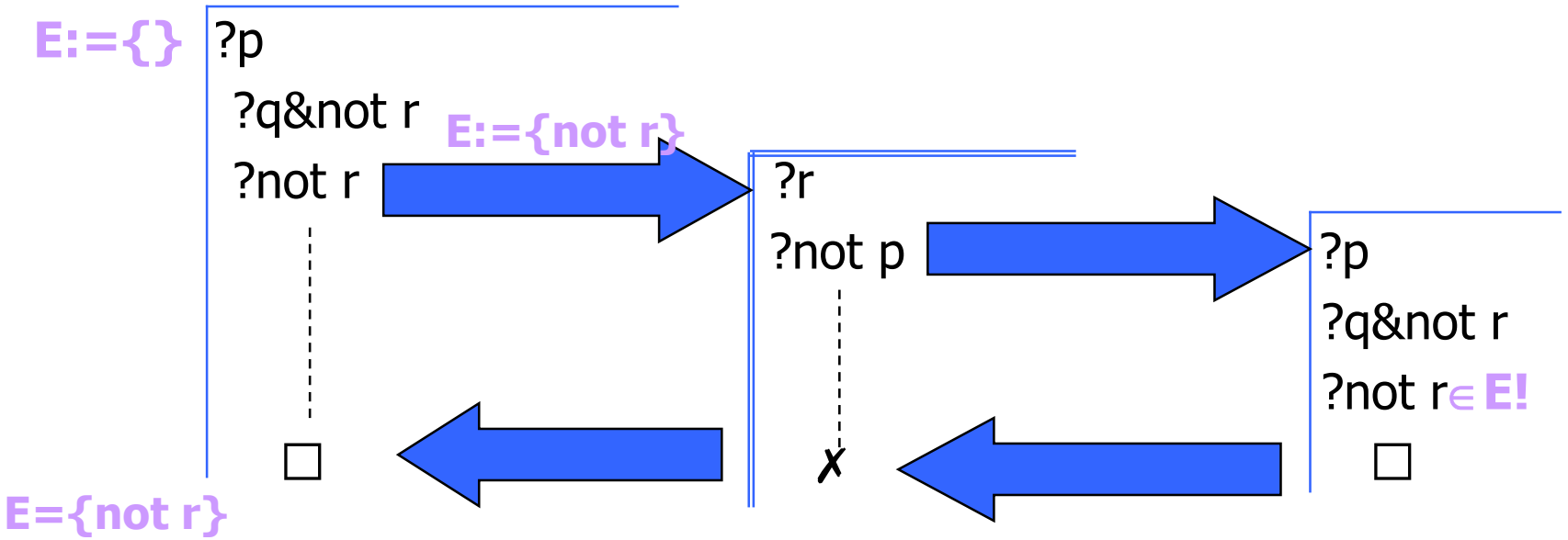
T: p **if** q **and not** r

q

r **if not** p

H={} IC={}

O: p



computed explanation is {}

Example (ASLDICN)

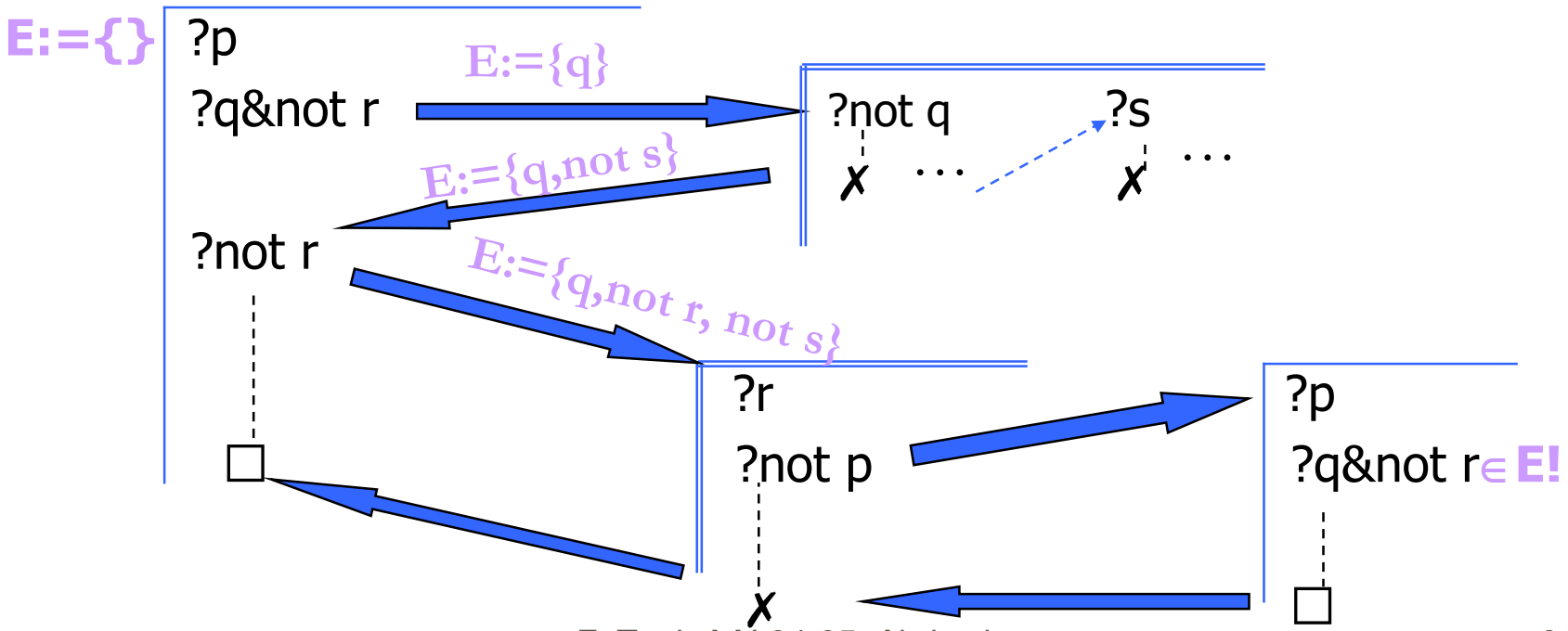
T: p **if** q **and not** r

r **if not** p

IC: **if** q **and** s **then false**

H: q,s

O: p



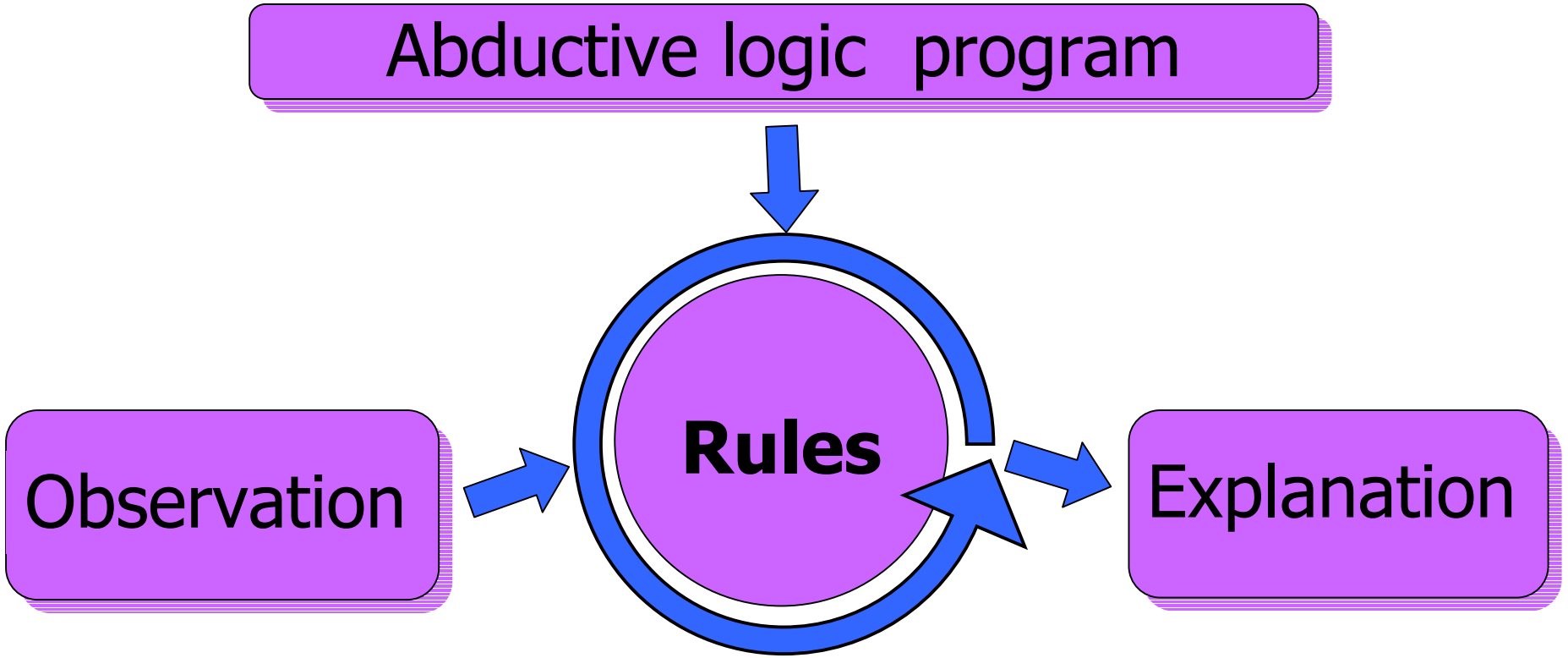
Abductive proof procedures (ASLD, ASLDIC, ASLDICN): limitations

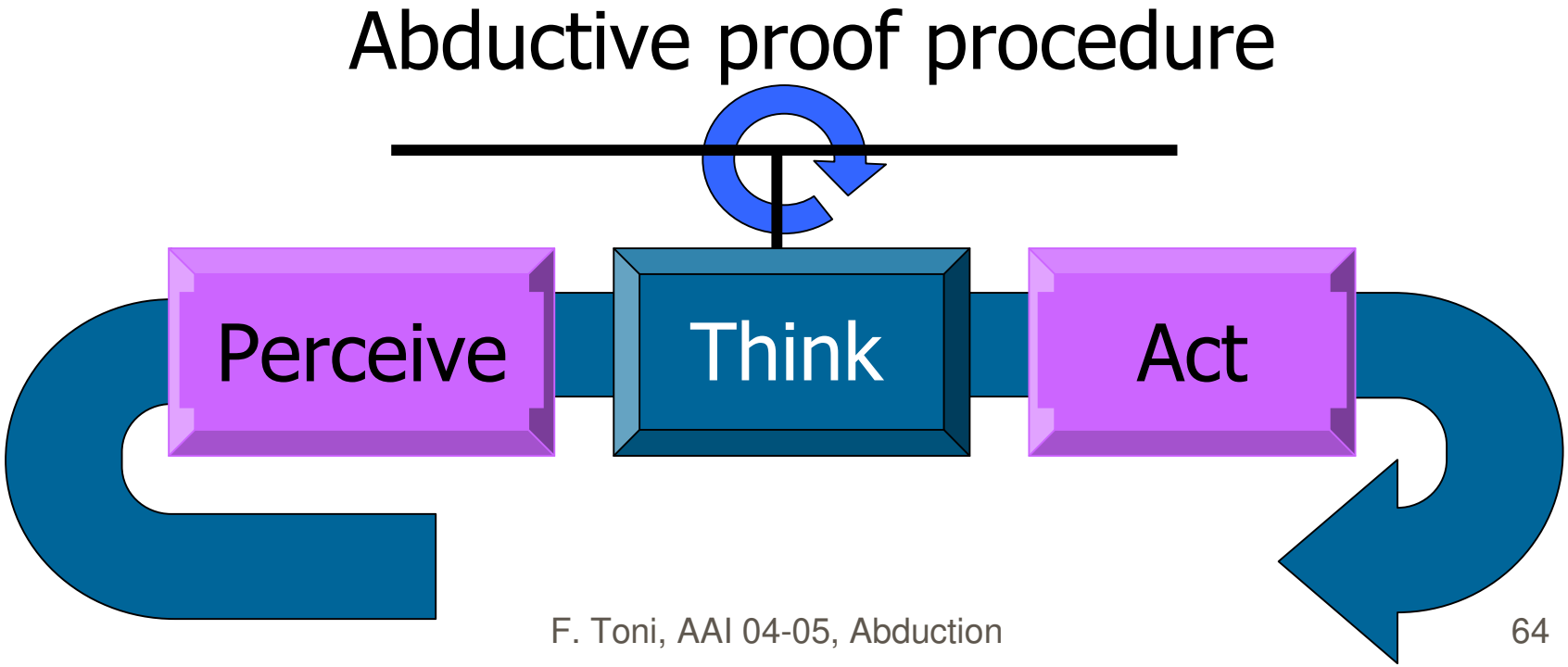
- *Safe* selection rule:
  - selected negative literals must be ground (**no constructive negation**)
  - selected atoms in abducible predicates must be ground (**no constructive abduction**)
- No proper treatment of constraint predicates ($<$, $\geq$)
- Restricted format for integrity constraints in abductive logic programs

CIFF: ALP with integrity constraints, constructive abduction and negation, dealing with constraint predicates

Execution of abductive proof procedures

- Static (off-line inputs-observations/outputs-explanations)
- ***Interactive*** and ***resource-bounded*** (on-line inputs/outputs)



Situated (rational/reactive) agents via ALP

- Kowalski,Sadri, *From Logic Programming to Multi-agent Systems*, Annals of Mathematics and AI, 1999
- Sadri,Toni,Torroni, *Resource allocation through negotiation via ALP*, Proc Jelia 2002
- Kakas,Mancarella,Sadri,Stathis,Toni, *The KGP model of agency,* Proc ECAI 2004