Research on Consequence Finding

Katsumi Inoue

National Institute of Informatics,

Tokyo, Japan

Franco-Japanese Meeting in Paris

September 2009

Contents

- Consequence finding problem (1967–1972)
- Restricted consequence finding (1990–1992)
- SOL Tableaux—SOLAR (2000–2009)
- Applications in AI (2001–2008)
  - answer extraction
  - agent systems, default reasoning
  - abduction
  - Induction (CF-induction)
  - scientific discovery (Systems Biology)
- Recent advances (2008–)
  - equality
  - subgoal decomposing
  - hypothesis evaluation (SOLAR+BDDEM)
  - meta-level abduction (Skill Science)

Consequence Finding

- Given an axiom set, the task of *consequence finding* or *theorem finding* is to find out some theorems of interest.
- Theorems to find out are not given in an explicit way, but are characterized by some properties.
- The task is clearly distinguished from *proof finding* or *theorem proving*.
- Theorem proving is a special case of consequence finding.

Consequence-finding Problem

<u>Resolution Principle</u>:

- refutation complete [Robinson, 1965]
- deductively incomplete
- Lee [1967]: **completeness theorem**

    Given a set of clauses ∑, for any clause D
    that is a logical consequence of ***∑,***
    RP can derive a clause C from **∑**
    such that <u>C entails/subsumes D</u>.

- Slagle, Chan & Lee [1969]: semantic resolution
- Minicozzi & Reiter [1972]: linear resolution

However, consequence finding has a problem …

The set of theorems is generally *infinite*, even if they are restricted to be minimal wrt subsumption.

⇓

[Siegel, 88], [Inoue, 90-92]

How to find only ***interesting* conclusions?**

Solutions: **Restricted Consequence Finding**

Production field and *characteristic clauses*

Production Field

- **Production field**: $\boldsymbol{P} = <\mathbf{L}, Cond>$
  - **L** : the set of literals to be collected
  - *Cond* : the condition to be satisfied (e.g. length)
- $Th_P(\Sigma)$: the clauses entailed by Σ which belong to $\boldsymbol{P}$.

□ $\boldsymbol{P1} = <\{ans\}^+,\ \text{none}>$:

- $\{ans\}^+$ is the set of positive literals with the predicate *ans*.
- $Th_{P1}(\Sigma)$ is the set of all positive clauses of the form $ans(t_1) \vee \ldots \vee ans(t_n)$ which are derivable from Σ.

□ $\boldsymbol{P2} = <\boldsymbol{L}^-,$ length is fewer than $k>$:

- $\boldsymbol{L}^-$ is the set of negative literals.
- $Th_{P2}(\Sigma)$ is the set of all negative clauses derivable from Σ consisting of fewer than k literals.

Characteristic Clauses

- **Characteristic clause** of Σ (wrt $\boldsymbol{P}$):

  A clause C such that

  – C belongs to $Th_{\boldsymbol{P}}(\Sigma)$;

  – no other clause in $Th_{\boldsymbol{P}}(\Sigma)$ subsumes C.

  ◆ $Carc(\Sigma, \boldsymbol{P}) = \mu Th_{\boldsymbol{P}}(\Sigma)$,

  where **μ** represents “subsumption-minimal”.

- **New characteristic clause** of C wrt Σ (and $\boldsymbol{P}$) :

  A char. clause of $\Sigma \wedge C$ which is not a char. clause of Σ.

  ◆ $NewCarc(\Sigma, C, \boldsymbol{P}) = \boldsymbol{\mu}[Th_{\boldsymbol{P}}(\Sigma \wedge C) - Th(\Sigma)]$

  $= Carc(\Sigma \wedge C, \boldsymbol{P}) - Carc(\Sigma, \boldsymbol{P})$.

Example: Group theory [Lee, 1967]

$$\Sigma = \{\ p(e,X,X),\quad p(i(X),X,e),$$
$$\neg p(X,Y,U) \vee \neg p(Y,Z,V)$$
$$\vee \neg p(U,Z,W) \vee p(X,V,W)\ \}$$

$$C = \neg p(X,Y,U) \vee \neg p(Y,Z,V)$$
$$\vee \neg p(X,V,W) \vee p(U,Z,V)$$

$$\boldsymbol{P} = \langle \{p\}^{+}, \text{length} \leqq 1 \text{ and term depth} \leqq 1 \rangle$$

$$N = \{\ \underline{p(X,i(X),e)},\ \underline{p(X,e,X)},\ p(e,e,i(e)),$$
$$p(i(X),X,i(e)),\ p(i(e),X,X),\ p(i(e),i(e),e)\ \}$$

Computing Characteristic Clauses

- *NewCarc(Σ,C,**P**)* (*C* : clause)

  can be directly realized by sound & complete consequence-finding procedures such as

  - **SOL resolution** [Inoue, 1992]
  - SFK resolution [del Val, 1999]

- *NewCarc(Σ,F,**P**)* (*F* : CNF formula)

  and *Carc(Σ, **P**)* can also be computed.

SOL Resolution [Inoue, 1991; 1992]

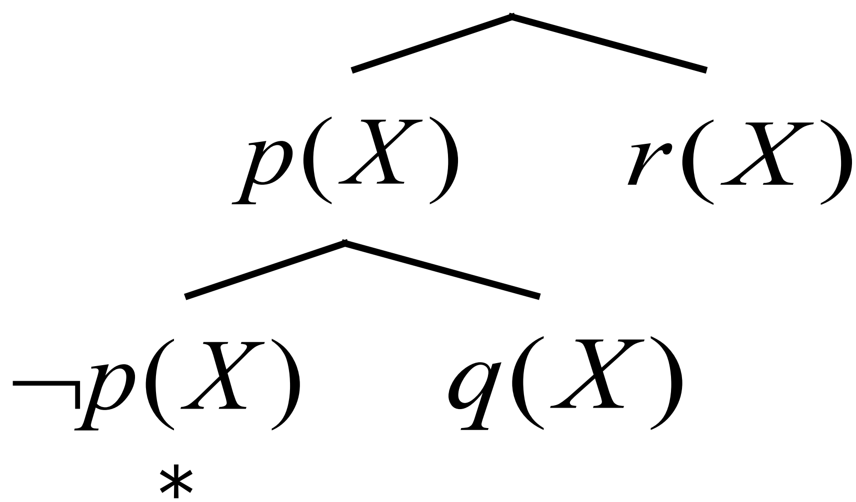
(Skipping Ordered Linear resolution)

- **Model Elimination + Skip rule**
- Skip, Resolve, Reduce rules
- complete for consequence-finding in *C-ordered linear resolution* (or ME) format
- complete for finding (*new*) *characteristic clauses*
- suitable for restricted consequence finding
- *connection tableau* format

  [Iwanuma, Inoue & Satoh, 2000]

Connection Tableau [Letz et al., 1994]

Clausal tableau whose every non-leaf node has an immediate successor labeled with the complementary literal.



SOL Resolution, Skip Rule

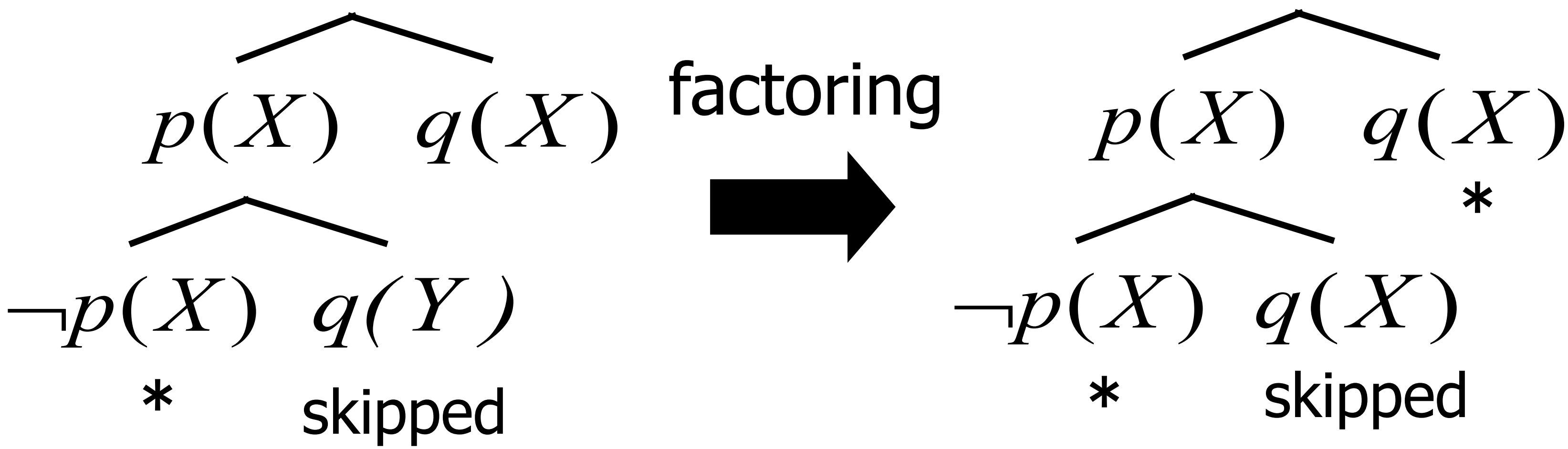
- **Skip** ･･･ When the selected literal *L* and already skipped literals belong to the production field, *L* is marked “skipped” and the branch is closed.

  Note: No substitution is applied.

  - When all branches in the tableau is closed, all the skipped literals represent a logical consequence that belongs to the production field.

SOL Resolution, Skip-factor Rule

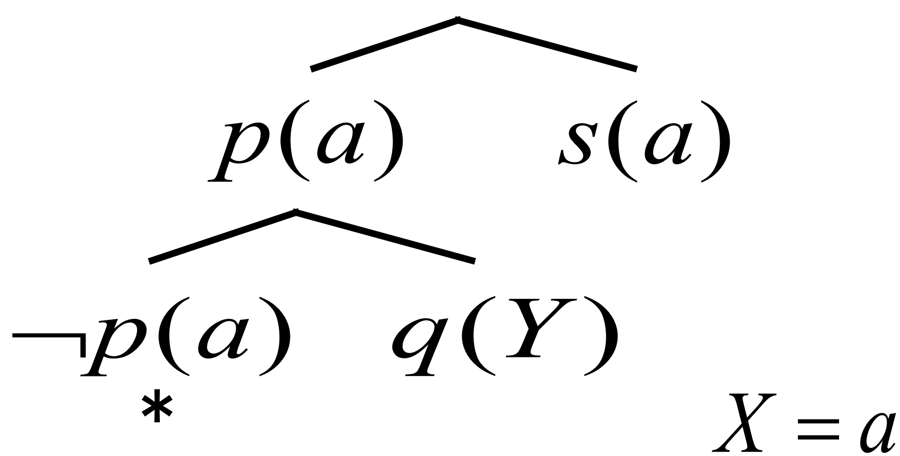
- **Reduce (factoring / merge)** ･･･When the selected literal *L* is unifiable with a leaf node in another branch, the branch is closed, and the substitution is applied.
This rule is only necessary for the skipped literals.



SOL Resolution, Resolve Rule

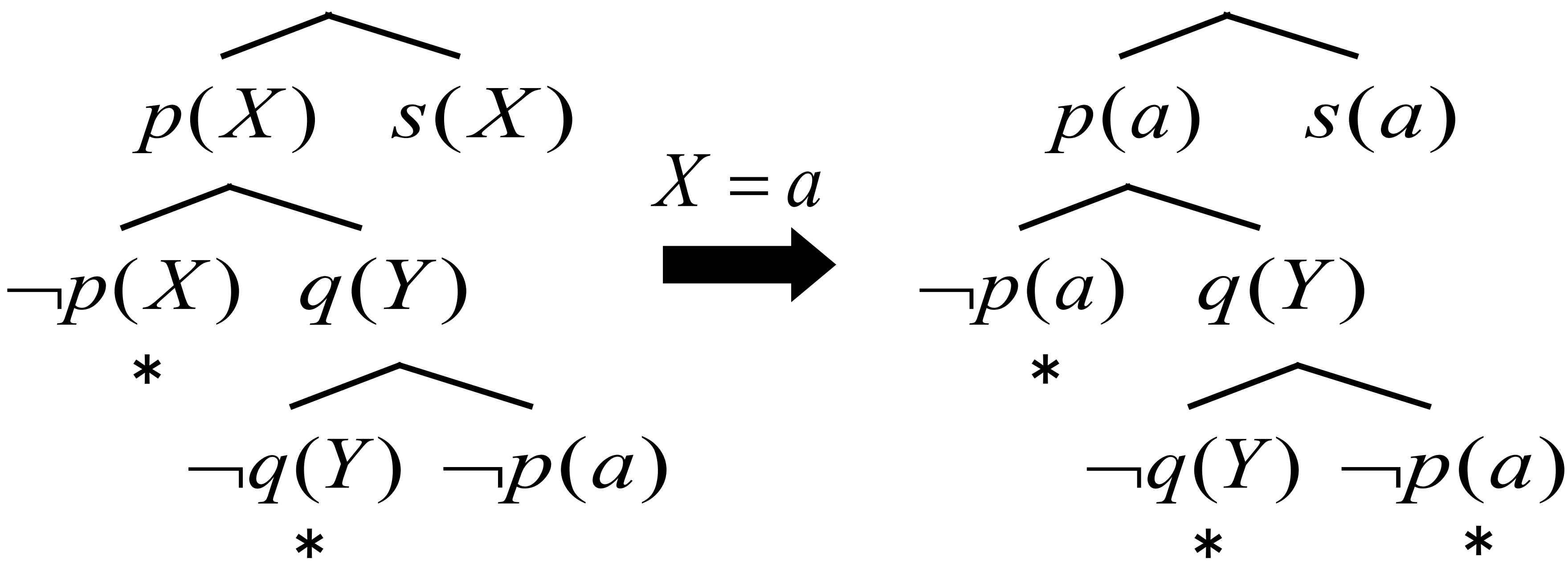
- **Resolve (extension)** ··· When the selected literal L is unifiable with the complement K of a literal in a clause from the axiom set, the clause is put under L, the branch with the complement K is closed, and the mgu substitution is applied in the whole tableau.

$$\frac{\begin{array}{c} p(X) \vee s(X) \\ \neg p(a) \vee q(Y) \end{array}}{q(Y) \vee s(a)}$$



SOL Resolution, Reduce Rule

- **Reduce (ancestry)** ･･･ When the selected literal *L* is unifiable with its ancestor, the branch is closed, and the mgu substitution is applied.



Example: New Characteristic Clauses

Axiom set: $\Sigma = \{\neg p(X) \vee q(X),\ \ \neg s(X),\ \ \neg p(X) \vee \neg q(X) \vee r(X)\}$

Input clause: $C = p(X) \vee s(X)$

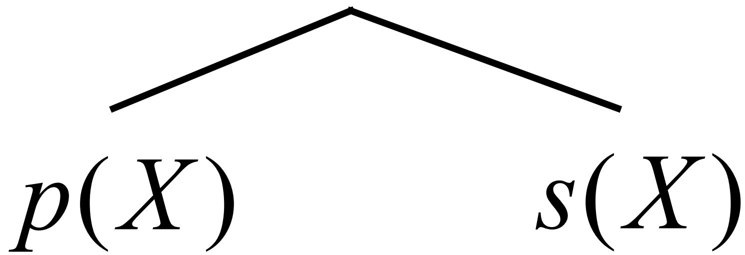
Production field: $\boldsymbol{P} = \{$positive literals, length≦2 $\}$

⬇

New characteristic clauses:

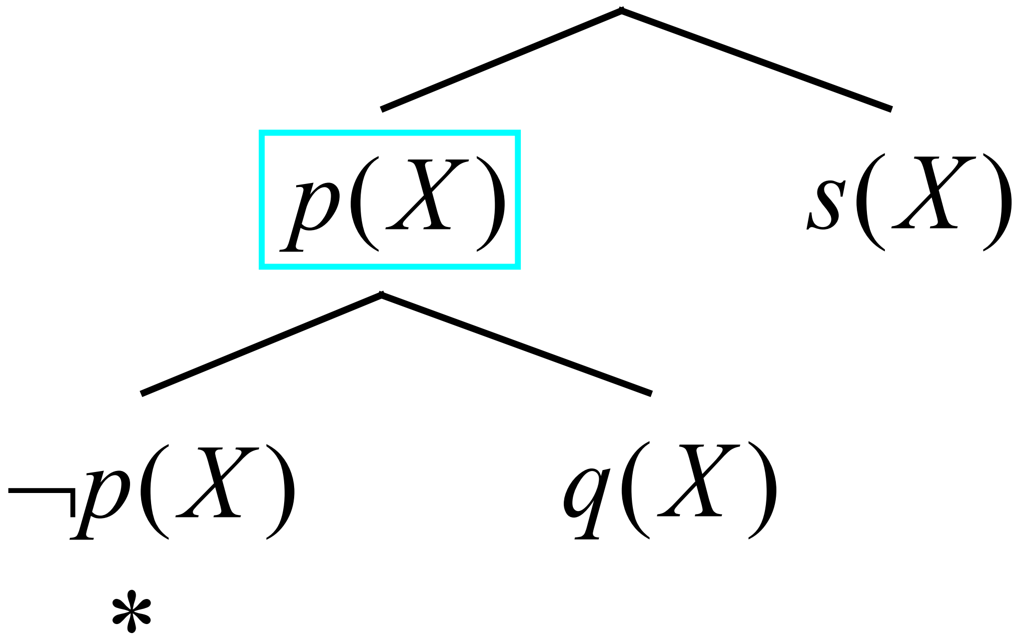
$$\boldsymbol{N} = \{p(X),\ \ q(X),\ \ r(X)\}$$

SOL Resolution, Example (1)



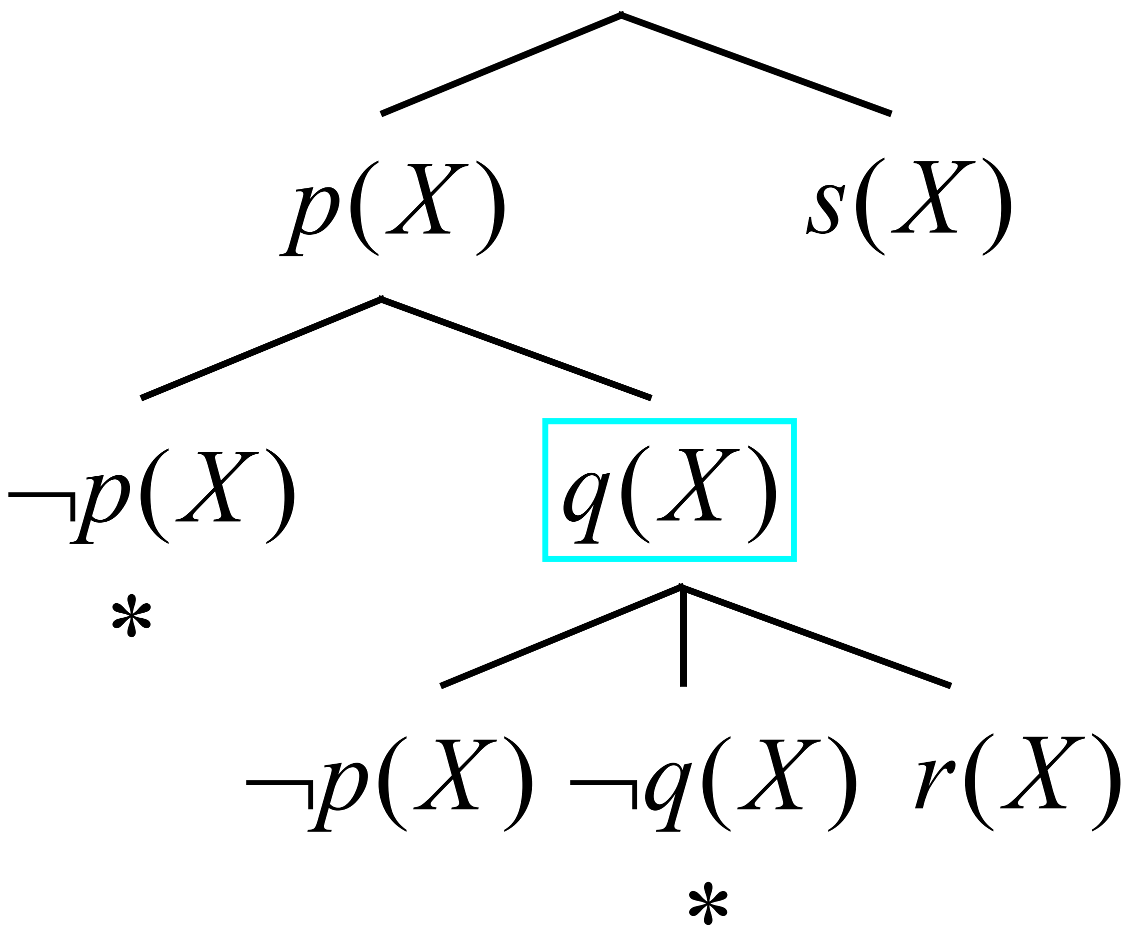
Start clause: $C = p(X) \vee s(X)$

SOL Resolution, Example (2)



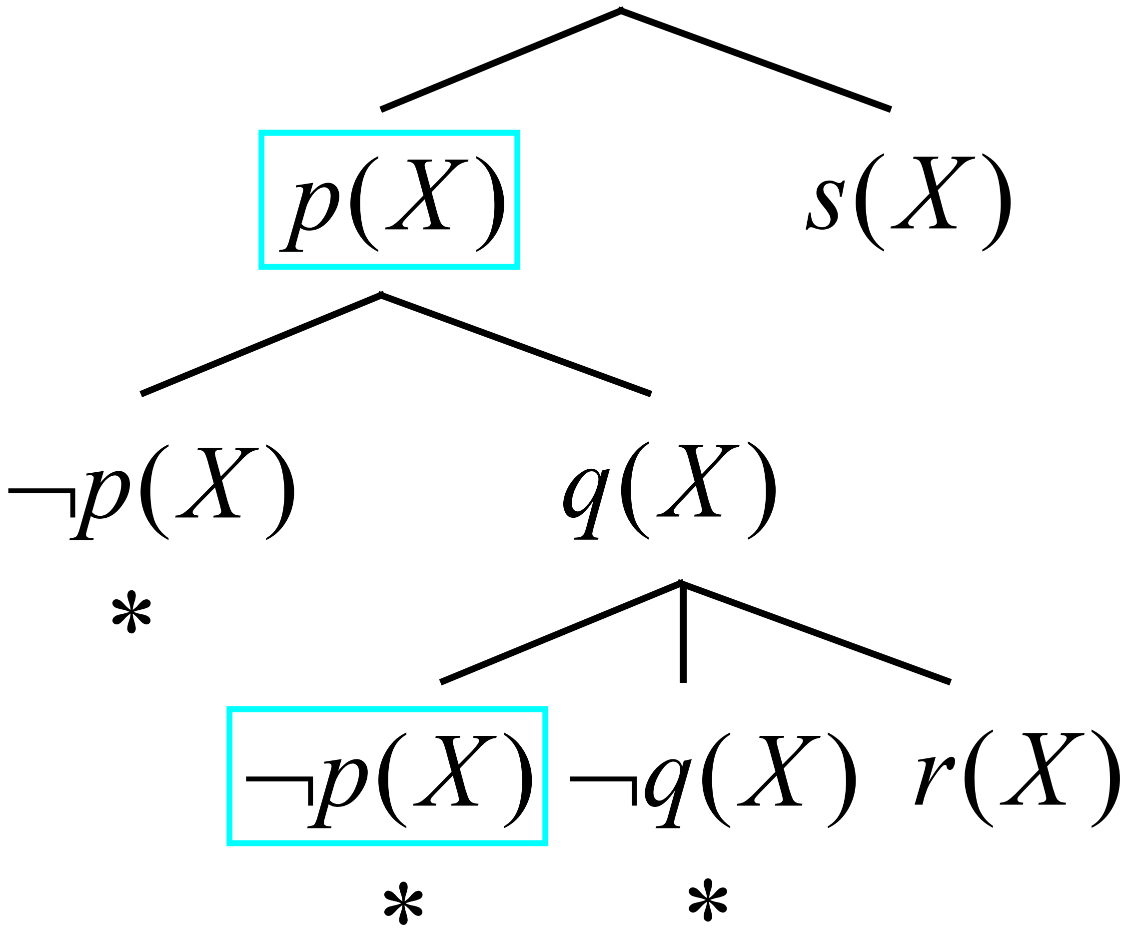
Resolve with $\neg p(X) \lor q(X)$

SOL Resolution, Example (3)



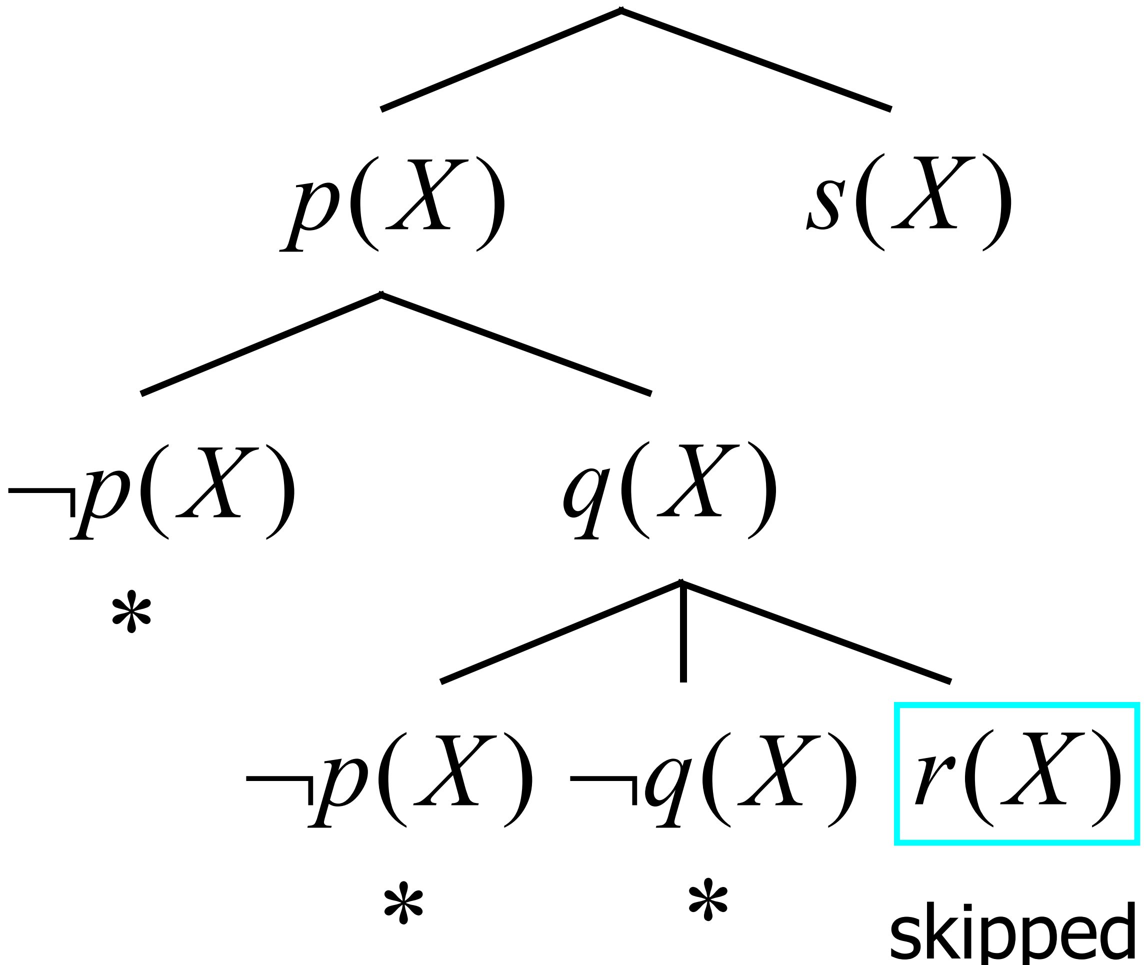
Resolve with $\neg p(X) \vee \neg q(X) \vee r(X)$

SOL Resolution, Example (4)



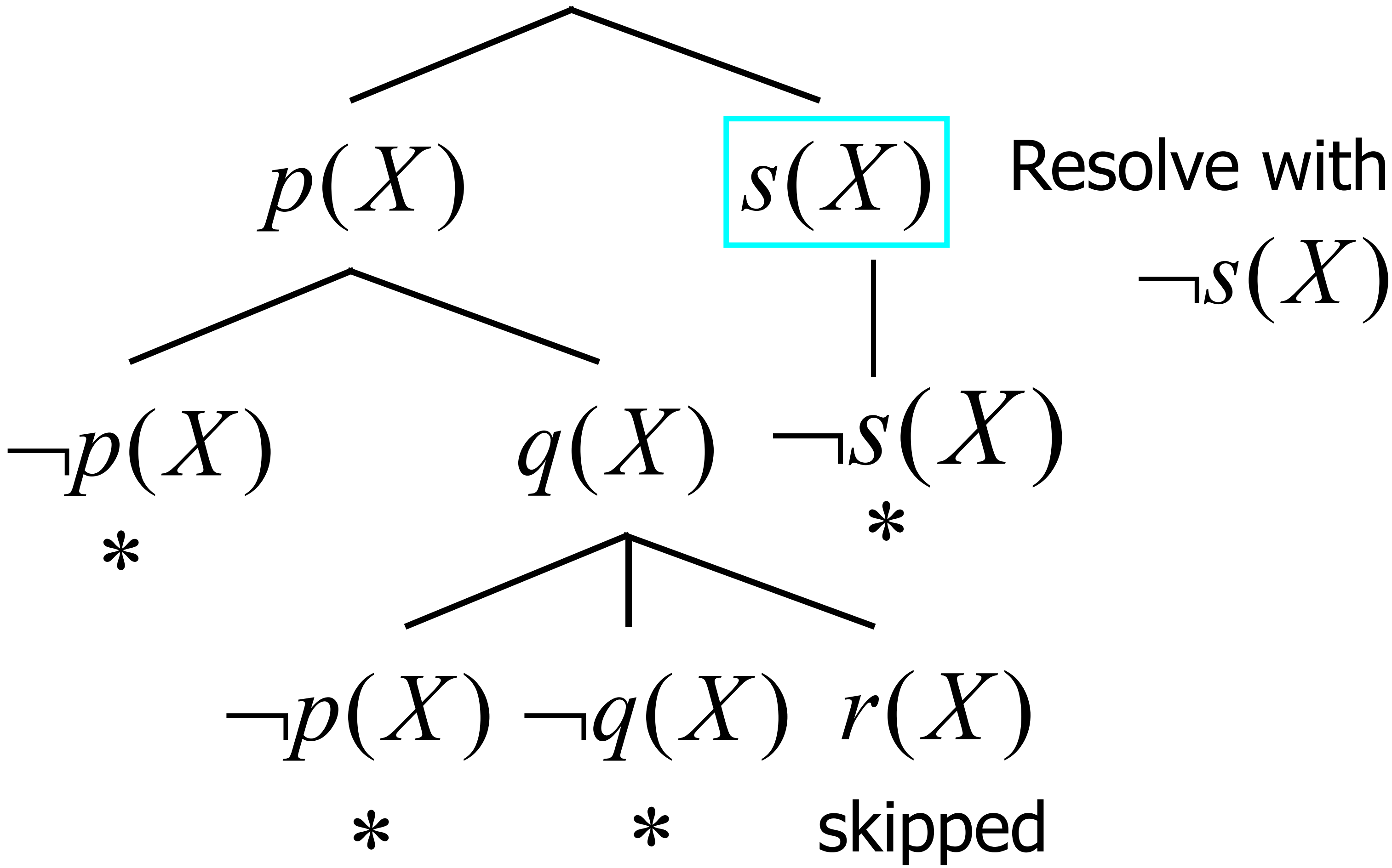
Reduce(ancestry)

SOL Resolution, Example (5)



Skipped $= \{r(X)\}$

SOL Resolution, Example (6)



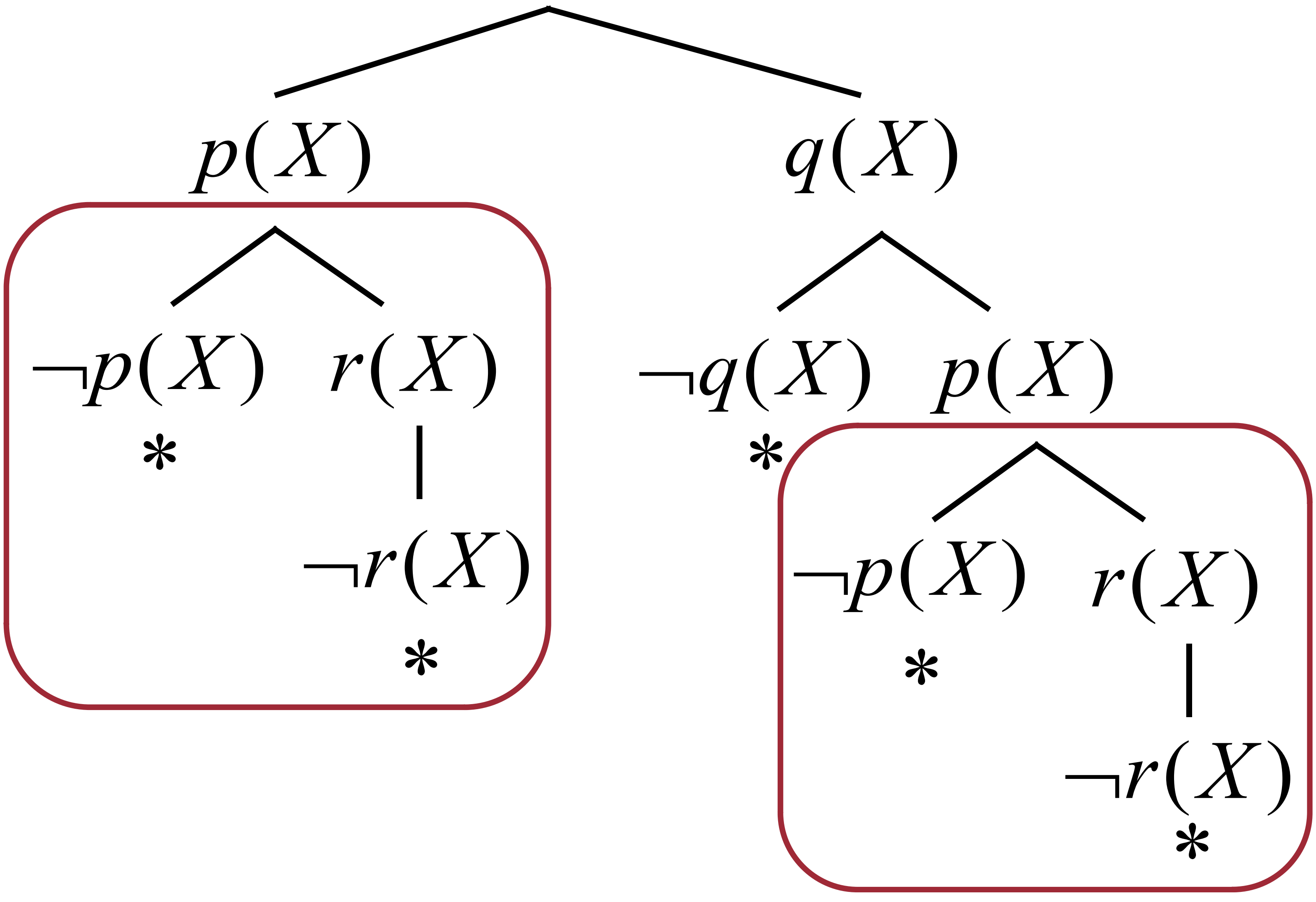
Skipped = $\{ r(X) \}$

Soundness and Completeness

1. If a clause *S* is derived by an SOL deduction from *Σ+C* and ***P***, then *S* belongs to *Th(Σ U{C})* and ***P***.

2. If a clause *F* does not belong to *Th(Σ)* but belongs to *Th(Σ U{C})* and ***P***, then there is an SOL deduction of a clause *S* from *Σ+C* and ***P*** such that <u>*S* subsumes *F*</u>.

Duplicated Computation



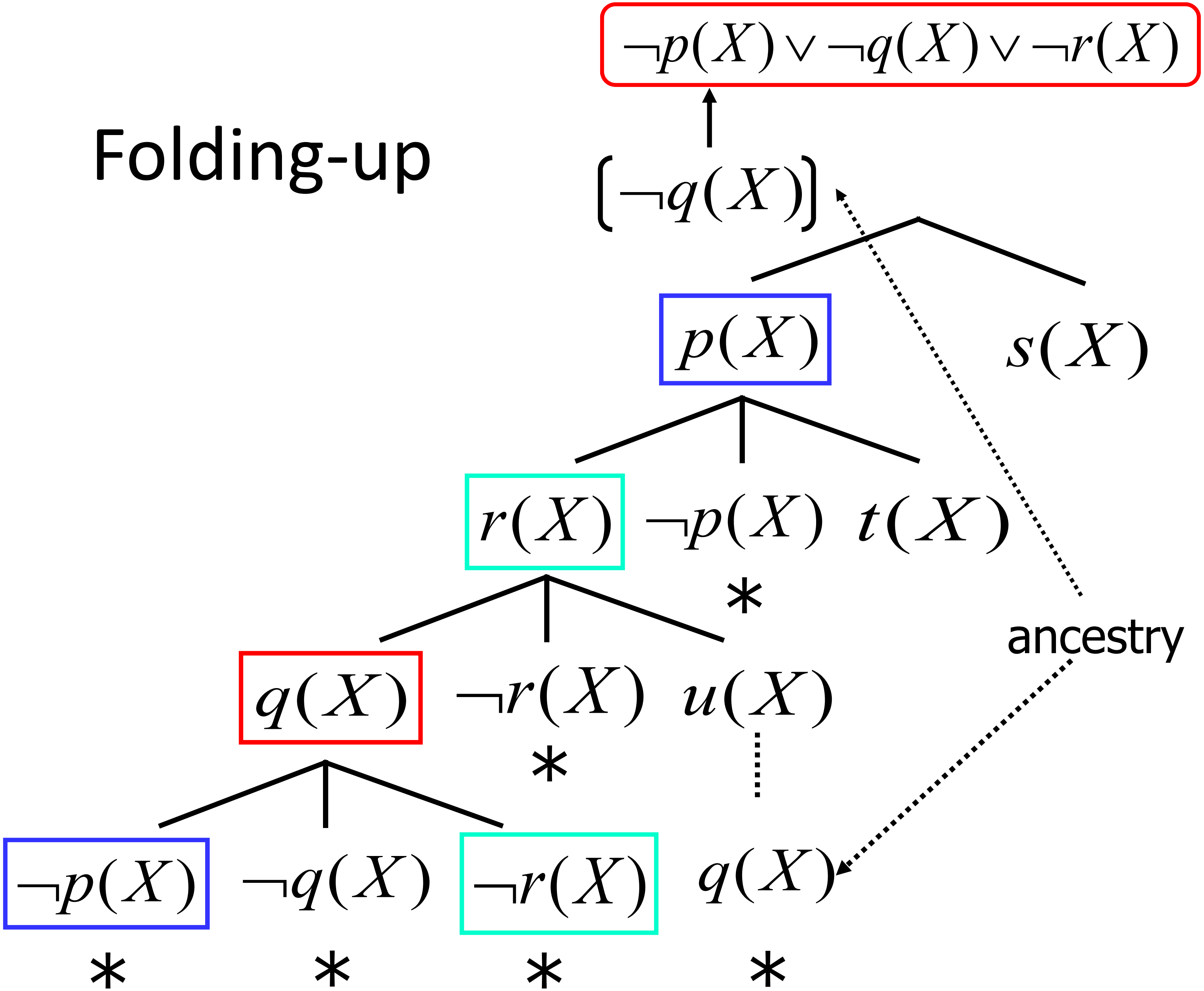
Pruning Methods in SOL Calculi

[Iwanuma, Inoue & Satoh, 2000]

- Mandatory Rules
  - lemma matching (unit/2-literals/folding-up)
  - merge with skipped literals
  - ancestry with empty substitution
  - C-reduction with empty substitution

- Cutting-off rules
  - regularity
  - tautology-freeness
  - complement-freeness
  - Skip as local failure pruning

Folding-up



SOLAR [Nabeshima, Iwanuma & Inoue, TABLEAUX 2003]

- Fast implementation of SOL Tableaux
- Java implementation
- Various pruning methods and constraints
- High performance as a theorem prover
  - Among 1,921 problems without the equality in TPTP v2.5.0, 52% Problems are solved by SOLAR within 5 min CPU time for each.
  - C.f. 50% are solved by OTTER 3.2 (C).

SOLAR 2.0

An efficient implementation of consequence finding procedure SOL (Nabeshima et al., 2008-2009)

- **Full checking of various pruning methods** [Iwanuma et al., 2000]
- Implementation based on **disequation constraints** [Letz & Stenz, 2001]
- **Term indexing mechanisms**
  - Perfect discrimination trees for term retrieval [McCune, 1992]
  - Feature vector indexing for clause-subsumption checking [Schulz, 2004]
- **Compact term data structure** with flat representation and variable offset
- **Non-recursive functions** with stacks which store the minimum essentials

Applications

- Nonmonotonic Reasoning
- Prime Implicants/Implicates, Knowledge Compilation
- Diagnosis, Design
- Problem Solving, Query answering, Planning
- Multi-Agent Systems
- Abduction
- Induction
- Scientific Discovery
- Skill Science

Flexible Query answering

- QA under incomplete information
- QA under incomplete communication environments
- QA in multi-agent systems

➢ We formalize FQA in logic:
  - Nonmonotonic reasoning
  - Default reasoning
  - Abductive reasoning

Communication under Incomplete Information

Under incomplete communication environments, *communication between agents is not guaranteed*. Messages between agents might be lost or delayed.

➢ [Satoh, Inoue, Iwanuma & Sakama, ICMAS-2000] proposed a method of *speculative computation* for reasoning/question-answering under incomplete communication environments in MAS.

Speculative Computation

[Satoh, Inoue, Iwanuma & Sakama, ICMAS 2000]

- Master-slave Multi-Agent System
- Master makes planning with **default answers** for slaves.
  - → Reduce suspended processes
  - → Reduce the risk
- When responses comes from slaves,
  - if the answer is the same as the default, keep the current computation process;
  - otherwise, recompute a plan.

SOL-based Speculative Computation

[Inoue, Kawaguchi & Haneda, CLIMA 2001]
[Iwanuma & Inoue, CLIMA 2002]
[Inoue & Iwanuma, AMAI 2004]

- Define a logical framework of MAS with speculative computation
  - **default logic** [Reiter, 80]
- Data-driven approach and bottom-up computation (*reactive* behavior)
  - **consequence-finding procedure (SOL)**
  - **avoidance of duplicate computation (History)**
- Implementation in a distributed environment with delayed inputs
  - **Servlet/Java-RMI and emails**

Query answering

Def : **program** $\sum$: a satisfiable set of clauses
query $\leftarrow Q$: a conjunction of literals

Def : Let θ_1, ..., θ_n be substitutions.

$Q\theta_1 \vee \ldots \vee Q\theta_n$ is a *correct answer* of $\sum$ if

$$\sum \models \forall \left(Q\theta_1 \vee \cdots \vee Q\theta_m \right)$$

Completeness of SOL for Computing Correct answers

Theorem : If $Q\theta_1 \vee \ldots \vee Q\theta_n$ is a correct answer of Σ, then there is an SOL deduction D from Σ s.t.

(1) the top clause is $\neg Q(X) \vee ans(X)$ [Green, 1969]

(2) the production field is $\boldsymbol{P} = \langle \{ans\}^+, \{\} \rangle$

(3) D derives a clause

$$ans(X)\delta_1 \vee \ldots \vee ans(X)\delta_k$$

which subsumes $ans(X)\theta_1 \vee \ldots \vee ans(X)\theta_n$.

Answer Completeness

[Iwanuma & Inoue, JELIA-2002]

- The completeness of SOL resolution implies the answer completeness.
- In particular, SOL resolution is complete for finding the minimal (length) answers.
- Currently, SOL is the only known complete calculus in the ME family.

c.f. P. Baumgartner, U. Furbach and F. Stolzenburg:

Computing answers with Model Elimination,

Artificial Intelligence, 90 (1997) pp.135-176.

Not all answers in condensed form can be computed.

Consequence Finding in Default Theories

Katsumi Inoue
National Institute of Informatics

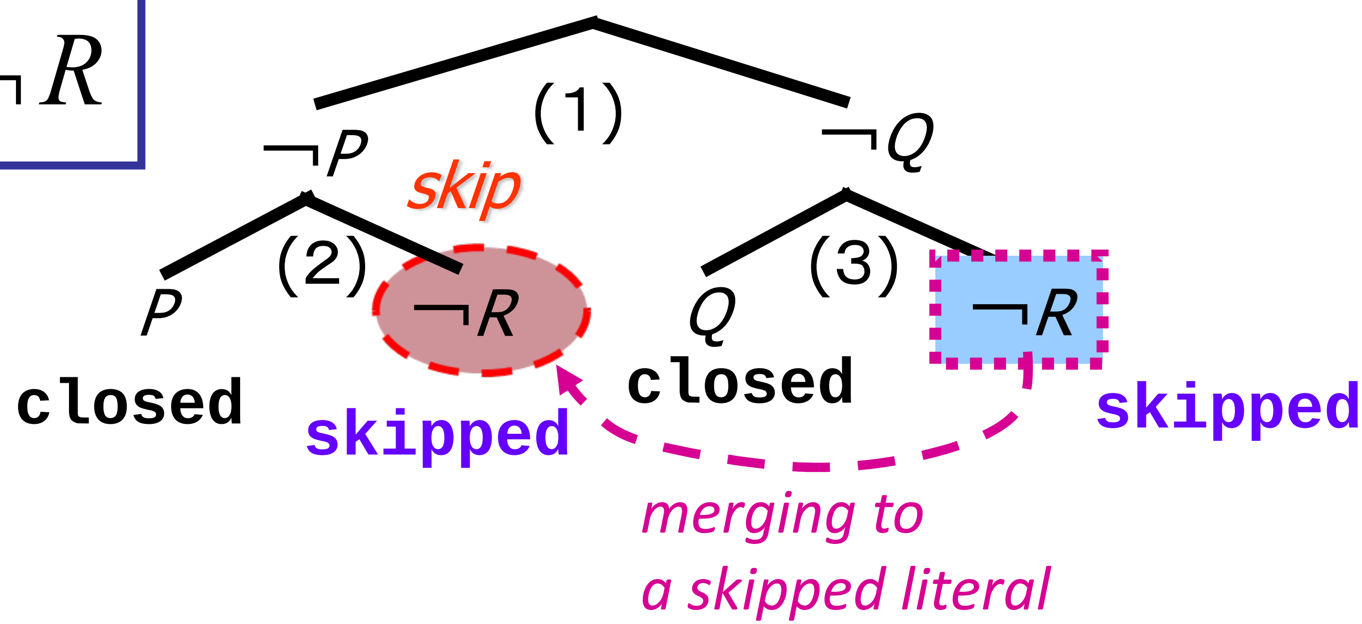
Koji Iwanuma
Hidetomo Nabeshima
University of Yamanashi

SOL Tableaux [Iwanuma, Inoue & Satoh, 00]: Connection Tableaux + Skip

Complete calculus for deriving logical consequences

Σ: (1) $\neg P \vee \neg Q$ (2) $P \vee \neg R$ (3) $Q \vee \neg R$

$$\Sigma \models \neg R$$



Default Theory

<u>prerequisite-free normal default theory</u> [Reiter, 80]

$\langle D, P \rangle$:

- D : *default set* : a set of ground literals.
- P : a set of first-order clauses, called a *program*, such that, if a clause in P contains a literal L whose predicate appears in D, L must be ground.

$\langle D, P \rangle$ corresponds to Reiter's default theory (D^*, P), where

$$\boldsymbol{D^*} = \left\{ \frac{:\ L}{L} \;\middle|\; L \in \boldsymbol{D} \right\}$$

Consequence-finding in Default Logic

- **<u>Theorem</u>** [Reiter, 87]:

  E is an **<u>extension</u>** of a default theory $\langle \boldsymbol{D}, \boldsymbol{P} \rangle$ <u>iff</u>

  $$E = Th\,(\,P \cup \Delta\,),$$

  where Δ is a maximal subset of $\boldsymbol{D}$ such that $P \cup \Delta$ is consistent.

- Δ is called the *generating defaults* for E.

➢ Extensions can be computed by consequence-finding from $P \cup \Delta$ if Δ can be computed in some way.

A Paper Review Problem

- There are 3 reviewers: #1, #2, #3.
- #2 and #3 usually accepts a paper, but #1 has no default.
- The editor asks each reviewer if the paper can be accepted.
- If all reviewers accepts the paper, it is ACCEPTED.
- If only 2 reviewers agrees to accept the paper, it is ACCEPTED WITH REVISONS.
- If neither #1 nor #2 accepts the paper, it is REJECTED, because they are key persons.

◆ Suppose that the editor gets a positive answer from #2 but no answers from #1 and #3 although the deadline has passed.

What should/can the editor decide in this situation?

Example: Paper Review

$\sum$ ={ *1, 2, 3*} : set of reviewer agents

D_1 = {*accept(2), accept(3)* }: defaults I % No default for #1

D_2 = {*accept(1)*, ¬*accept(1)*, *accept(2), accept(3)* }: defaults II

% Two defaults for #2 which are complementary

P : program

¬*accept*(*1*) ∨ ¬*accept*(*2*) ∨ ¬*accept*(*3*) ∨ *rank*(*A*, [*1,2,3*]).

¬*accept*(*1*) ∨ ¬*accept*(*2*) ∨ *accept*(*3*) ∨ *rank*(*B*, [*1,2*]).

accept(*1*) ∨ ¬*accept*(*2*) ∨ ¬*accept*(*3*) ∨ *rank*(*B*, [*2,3*]).

¬*accept*(*1*) ∨ *accept*(*2*) ∨ ¬*accept*(*3*) ∨ *rank*(*B*, [*1,3*]).

accept(*1*) ∨ *accept*(*2*) ∨ *rank*(*C*, []).

Example: Paper Review

$D_1 = \{a(2), a(3)\}$: defaults I

$D_2 = \{a(1), \neg a(1), a(2), a(3)\}$: defaults II

P : program

$$\neg a(1) \vee \neg a(2) \vee \neg a(3) \vee r(A, [1,2,3]).$$
$$\neg a(1) \vee \neg a(2) \vee a(3) \vee r(B, [1,2]).$$
$$a(1) \vee \neg a(2) \vee \neg a(3) \vee r(B, [2,3]).$$
$$\neg a(1) \vee a(2) \vee \neg a(3) \vee r(B, [1,3]).$$
$$a(1) \vee a(2) \vee r(C, []).$$

- $\langle D_1, P \rangle$ has 1 extension, but $\langle D_2, P \rangle$ has 2 extensions.
- $r(a, [1,2,3]) \vee r(b, [2,3])$ is a consequence of both $\langle D_1, P \rangle$ and $\langle D_2, P \rangle$.

1st Step: Consequence-finding with Defaults in SOL with answer literals

Theorem: Suppose that Δ is a maximal subset of D such that $P \cup \Delta$ is consistent. If $Q(X)\theta_1 \vee \ldots \vee Q(X)\theta_n$ is a correct answer to the query $\leftarrow Q(X)$ relative to $P \cup \Delta$,

then there is an SOL-deduction S from $P \cup \Delta$ such that:

(1) the top clause is $\neg Q(X) \vee ans(X)$.

(2) the production field is $\mathcal{P} = \langle \{ans\}^+, \{\} \rangle$.

(3) S generates a clause $ans(X)\sigma_1 \vee \ldots \vee ans(X)\sigma_k$ which subsumes $ans(X)\theta_1 \vee \ldots \vee ans(X)\theta_n$.

Note: Δ must be computed in advance.

Conditional answers

- *Query* $\leftarrow Q(X)$: $Q(X)$ is a conjunction of literals

- *Conditional answer* to $\leftarrow Q(X)$ relative to ***Σ*** and ***P*** :

  A clause of the form:

  $$A_1 \vee \ldots \vee A_m \vee Q(X)\theta_1 \vee \ldots \vee Q(X)\theta_n$$

  s.t. (1) $\boldsymbol{\Sigma} \models A_1 \vee \ldots \vee A_m \vee Q(X)\theta_1 \vee \ldots \vee Q(X)\theta_n$,

  (2) $\boldsymbol{\Sigma} \not\models A_1 \vee \ldots \vee A_m$, (3) $A_1 \vee \ldots \vee A_m$ belongs to ***P*** .

- *Conditional ans-clause (CA-clause)* relative to ***Σ*** and ***P*** :

  A clause in the form of

  $$A_1 \vee \ldots \vee A_m \vee ans(X)\theta_1 \vee \ldots \vee ans(X)\theta_n$$

  s.t. $A_1 \vee \ldots \vee A_m$ satisfies the same 3 conditions as above.

Computing Default Consequences as Conditional answers

Conditional answer format explicitly represents:

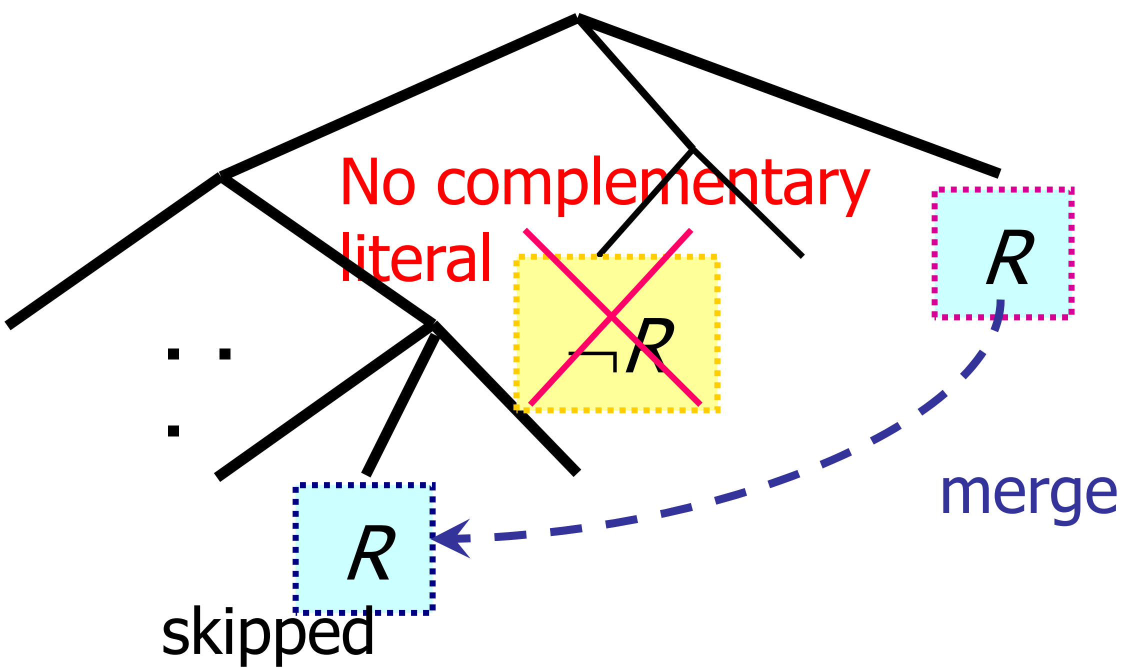
which defaults are used to derive the conclusion.

The dependency representation is valuable for avoiding duplicated computations when there are multiple extensions sharing common defaults.

- SOL tableaux can reduce redundant computation which derives irrational conclusions in the conditional answer format by means of the skip-regularity and TCS-freeness constraints.

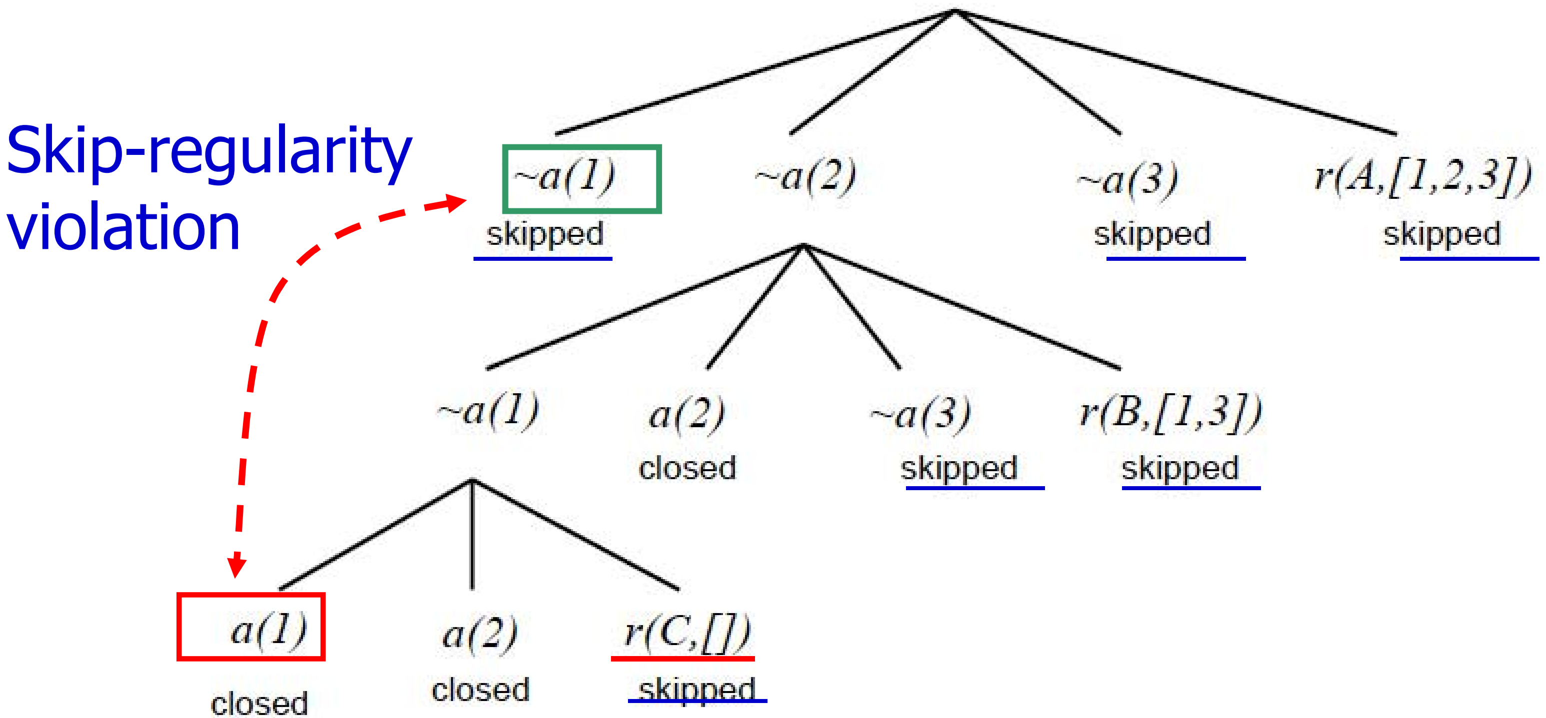
Constraint: Skip-Regularity

Any *complementary literals* of skipped literals can be forbidden to appear in an SOL tableau, without losing the completeness.



Irrational answers Violating Skip-Regularity

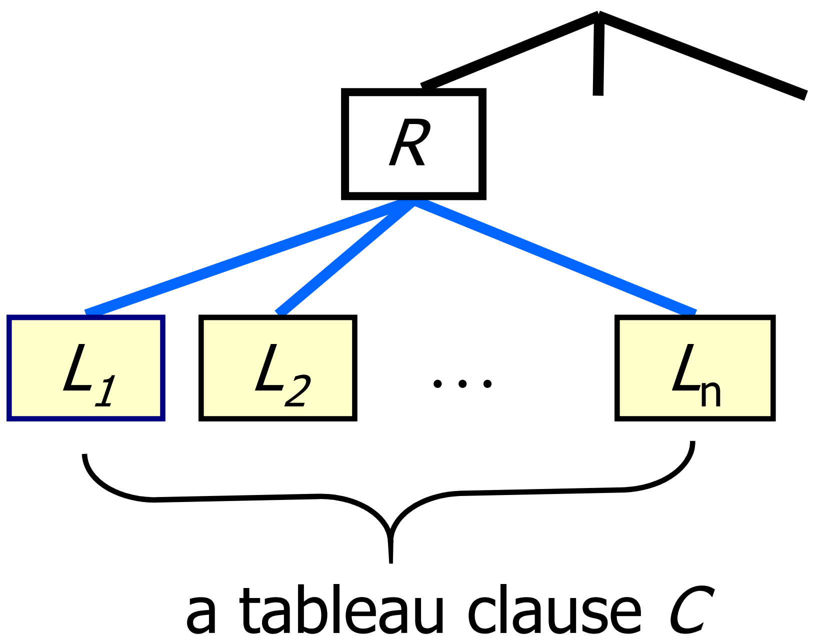
The tableau violates the skip-regularity wrt $a(1)$.



$$a(1) \wedge a(3) \rightarrow r(A,[1,2,3]) \vee r(B,[1,3]) \vee r(C,[])$$

Constraint: TCS (Tableau Clause Subsumption)-Freeness

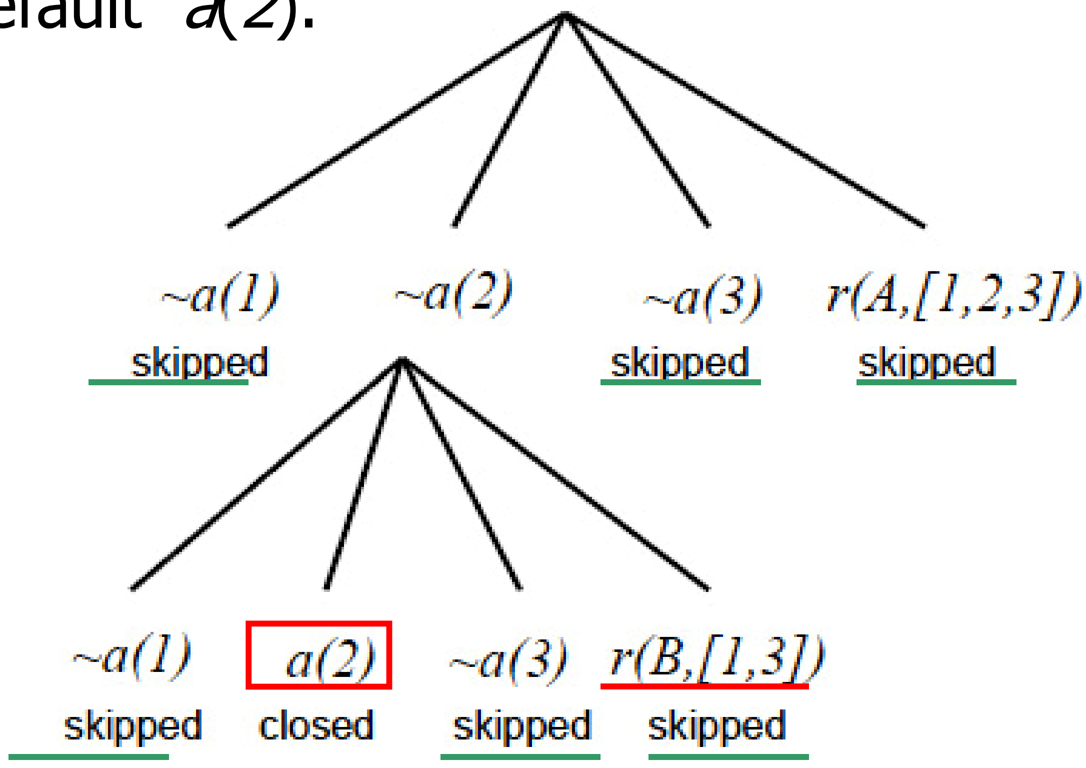
Any tableau clause *C* (a disjunction of sibling literals in a tableau) is not subsumed by any clause in *Σ* other than origin clauses of *C*.



Σ: a set of clauses as an axiom theory

Irrational answers Violating TCS-Freeness

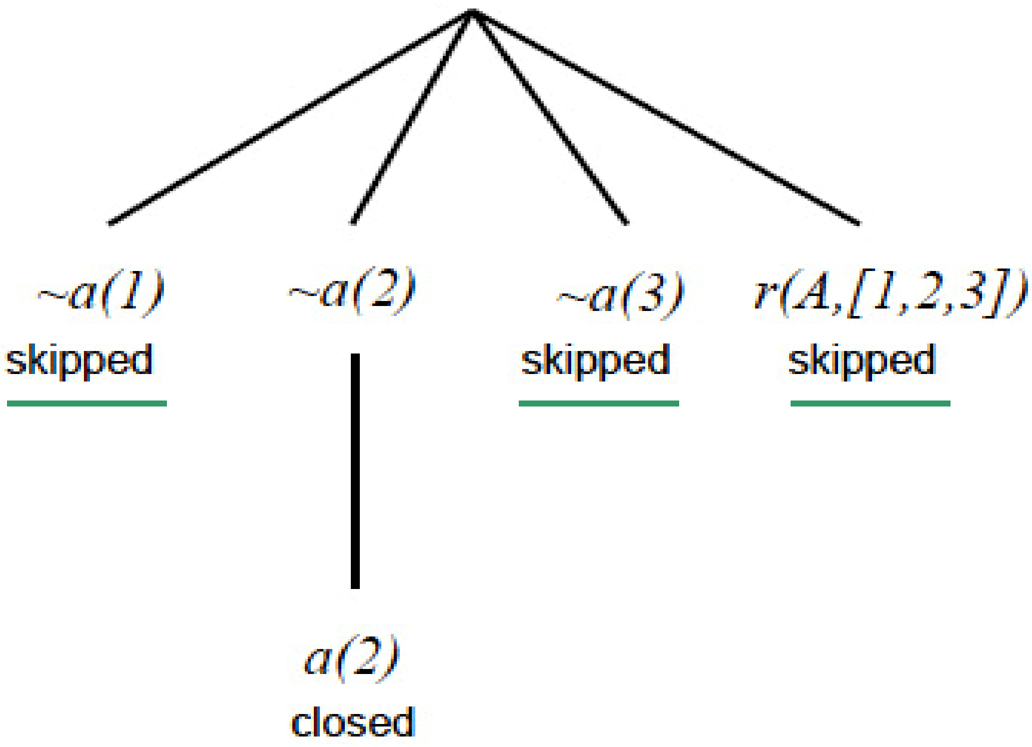
The lower tableau clause is subsumed by newly added default $a(2)$.



Skip-regular but not TCS-free for the new underlying theory

$$a(1) \wedge a(3) \rightarrow r(A,[1,2,3]) \vee r(B,[1,3])$$

Rational answers Satisfying Skip-Regularity and TCS-Freeness



$$a(1) \wedge a(3) \rightarrow r(A,[1,2,3])$$

2nd step: Consequence-finding with Defaults in Conditional answer Format

Theorem: Suppose that Δ is a maximal subset of D such that $P \cup \Delta$ is consistent. If $Q(X)\theta_1 \vee \ldots \vee Q(X)\theta_n$ is a correct answer to the query $\leftarrow Q(X)$ relative to $P \cup \Delta$, then there is an SOL-deduction S from P such that:

(1) the top clause is $\neg Q(X) \vee ans(X)$.

(2) the production field is $P = \langle D^{-} \cup \{ans\}^{+}, \{\} \rangle$.

(3) S generates a CA-clause of the form

$$B_1 \vee \ldots \vee B_s \vee ans(X)\sigma_1 \vee \ldots \vee ans(X)\sigma_k :$$

- each B_i is the negation of a default in D.
- $ans(X)\sigma_1 \vee \ldots \vee ans(X)\sigma_k$ subsumes $ans(X)\theta_1 \vee \ldots \vee ans(X)\theta_n$.

Problems Unsolved Yet

Exclusion of the generating defaults from the axiom set implies that these literals cannot be regarded as unit clauses that are newly added to the axiom set.

1. Resolve with default literals becomes impossible.

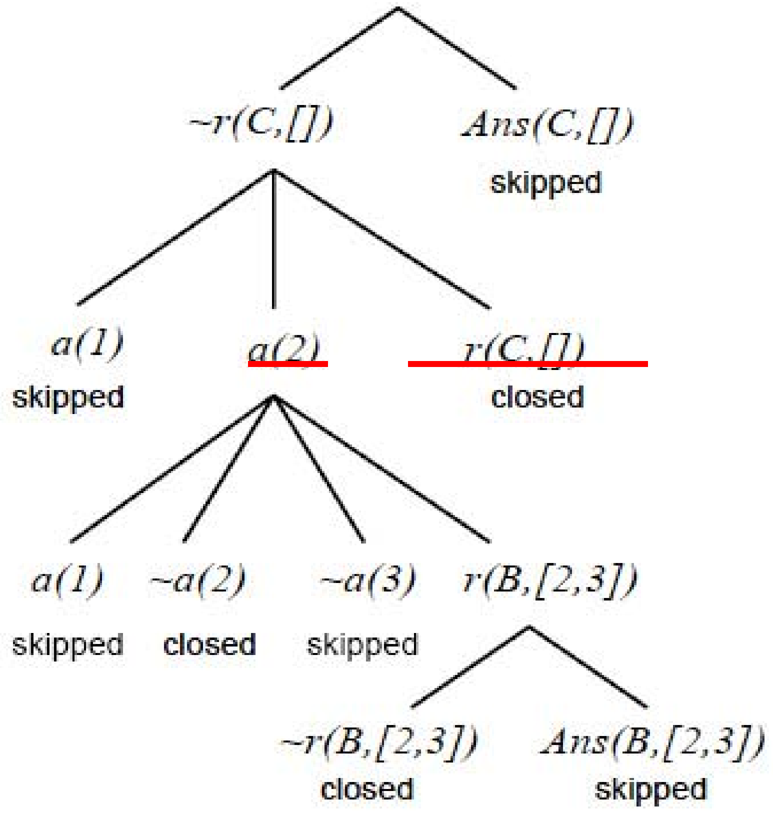
   ⇒ Skip-preference rule

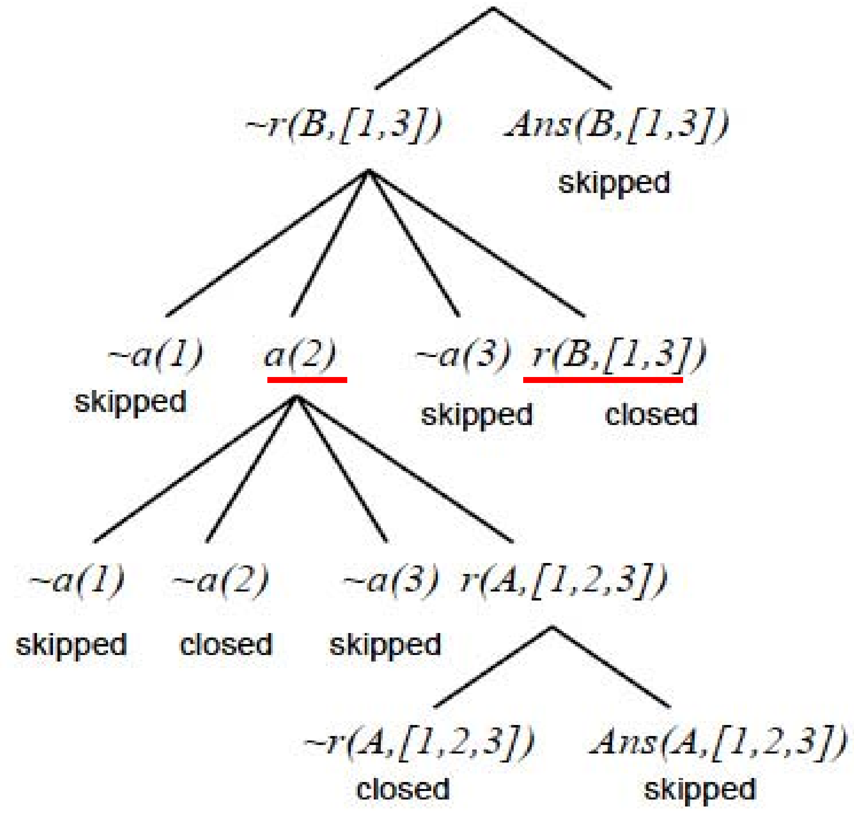
2. TCS-freeness constraint by default literals becomes inapplicable to tableaux. Then, many irrational tableaux cannot be pruned.

   ⇒ Γ-subsumption rule

Irrational Tableaux Example

Default literal: $a(2)$.





$$\neg a(1) \wedge a(3) \rightarrow ans(C,[]) \vee ans(B,[2,3])$$

$$a(1) \wedge a(3) \rightarrow ans(A,[1,2,3]) \vee ans(B,[1,3])$$

SOL-S(Γ) calculus:

SOL + *Skip-preference* + *Γ-subsumption*

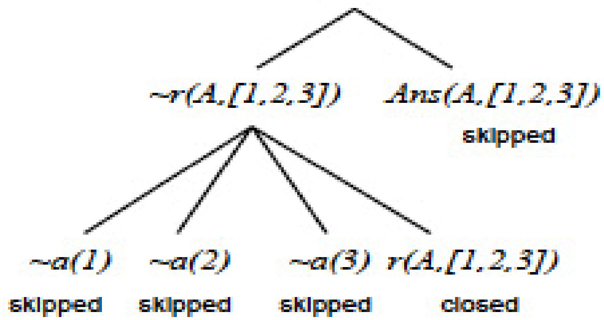
1. Skip-preference:

   Apply Skip as much as possible by ignoring the possibility of other inference rules. The extension (Resolve) with default literals in Γ can completely be simulated.

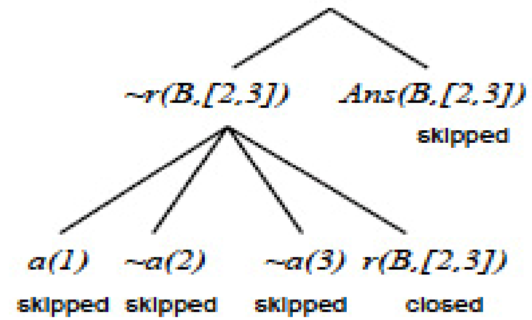
2. Γ-subsumption checking:

   Check if a selected subgoal belongs to a set Γ of default literals, and if so the tableau is pruned.
   This check cannot simulate the complete TCS-subsumption, but is enough for consequence-finding with defaults.

4 Survived Rational Tableaux in SOL-S(Γ) (from 3,184 original SOL Tableaux)

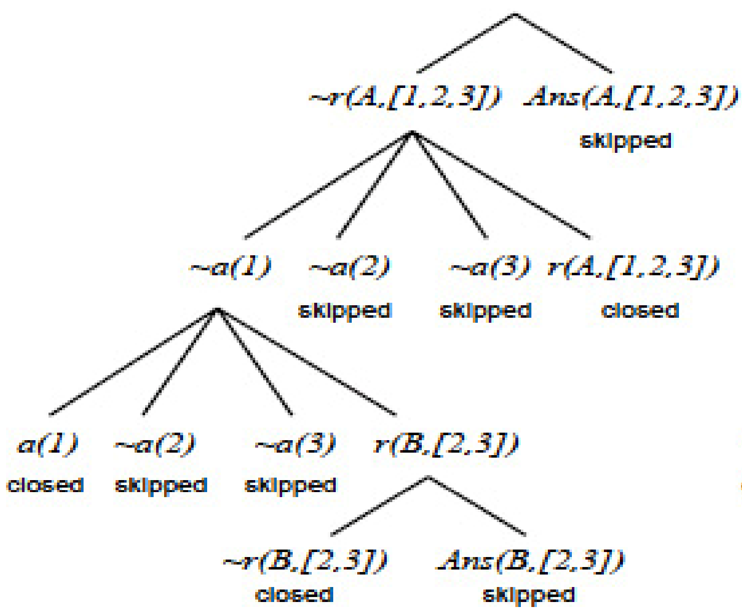


(a)

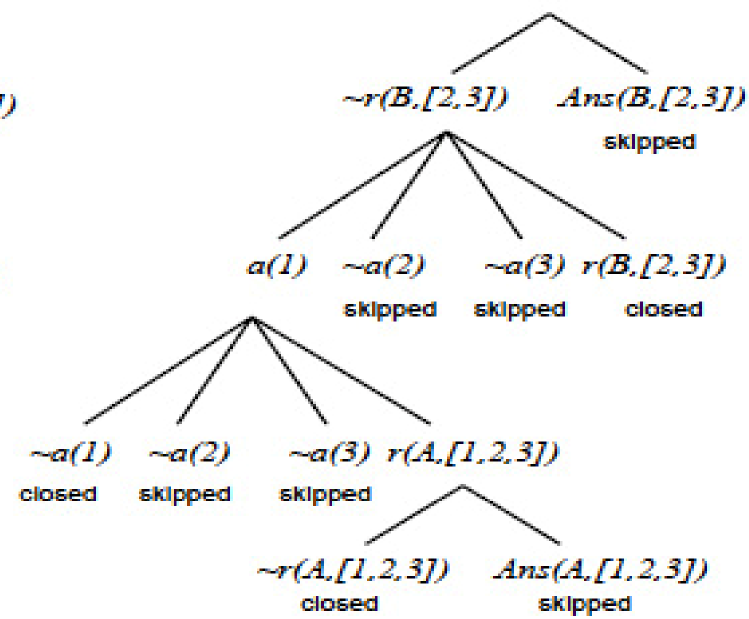


(b)

$\Gamma = \{ \neg a(2), \neg a(3) \}$



(c)



(d)

3rd step: Consequence-finding with Defaults in SOL-S(Γ) calculus

Theorem: Suppose that Δ is a maximal subset of D such that $P \cup \Delta$ is consistent. If $Q(X)\theta_1 \vee \ldots \vee Q(X)\theta_n$ is a correct answer to the query $\leftarrow Q(X)$ relative to $P \cup \Delta$, and

$\Gamma = \{ \neg L \in D^- \mid \neg L$ appears nowhere in $Card(P, < D^-, \{\}>)\}$,

then there is an SOL-S(Γ) deduction S from P such that:

(1) the top clause is $\neg Q(X) \vee ans(X)$.

(2) the production field is $\mathcal{P} = < D^- \cup \{ans\}^+, \{\}>$.

(3) S generates a CA-clause of the form

$$B_1 \vee \ldots \vee B_s \vee ans(X)\sigma_1 \vee \ldots \vee ans(X)\sigma_k :$$

- each B_i is the negation of a default in D.
- $ans(X)\sigma_1 \vee \ldots \vee ans(X)\sigma_k$ subsumes $ans(X)\theta_1 \vee \ldots \vee ans(X)\theta_n$.

Experimental Result

	SOL Deductions	# of Char. Clauses	Time (ms)
Original SOL	3184	3	8225
SOL with Skip-Regul.	1270	3	2937
SOL-S(Γ)	4	3	34

4th step: Consistency of SOL(-S) calculus for consequence-finding with defaults

Theorem: Let (D,P) be a default theory, and $\leftarrow Q(X)$ a query.

If there is an SOL(-S(Γ)) deduction S from P such that

(1) the top clause is $\neg Q(X) \vee ans(X)$.

(2) the production field is $P = < D^{-} \cup \{ans\}^{+}, \{\}>$.

(3) S generates $B_1 \vee \ldots \vee B_s \vee ans(X)\sigma_1 \vee \ldots \vee ans(X)\sigma_k$:

- each B_i is the negation of a default in D.
- $B_1 \vee \ldots \vee B_s$ is not subsumed by any clause in $Card(P, <D^{-}, \{\}>)$,

then there is an extension E of (D,P) such that

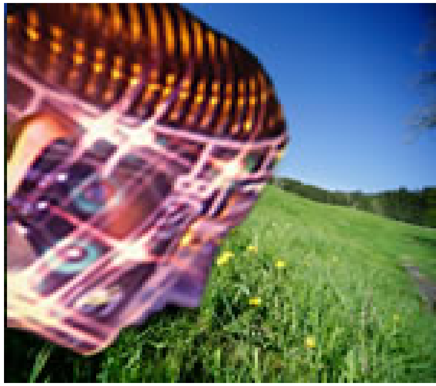
- $\{\neg B_1, \ldots, \neg B_s\} \subseteq \Delta$, where Δ is the generating defaults of E.
- E contains the correct answer $Q(X)\sigma_1 \vee \ldots \vee Q(X)\sigma_n$.

Summary

- Consequence-finding in default logic is considered.
- Default computation is verified by consequence-finding from the axioms with the generating defaults.
- A sound and complete answer extraction technique can be provided with SOL tableaux.
- Conditional answer format is useful for representing dependencies between consequences and defaults, thereby providing consequence-finding without computing the generating defaults.
- Skip-preference and Γ-subsumption prevents generating irrational consequences under defaults.
- The framework can be applied to speculative computation in multi-agent systems.

Three modes of inference

(C.S. Peirce)



deduction

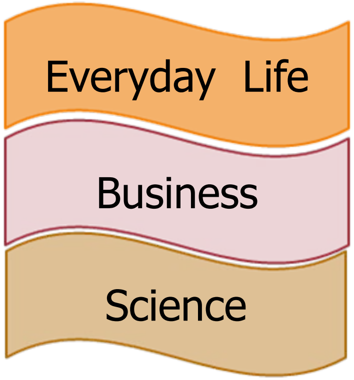
$$\frac{A \rightarrow C \qquad A}{C}$$

induction

$$\frac{A \qquad C}{A \rightarrow C}$$

abduction

$$\frac{A \rightarrow C \qquad C}{A}$$



Abduction and Induction: Logical Framework

Input:

- B : background theory
- E : (positive) examples ／observations

Output:

- H : hypothesis satisfying that
  - $B \wedge H \models E$
  - $B \wedge H$ is consistent.

Abduction and Induction: Logical Framework

- $B \wedge H \models E$
- $B \wedge H$ is consistent.

- The logical framework is exactly the same.
- A different formalism exists for induction, e.g., *descriptive induction*, but can be unified with the above framework [Inoue & Saito, ILP'04].
- Induction often gets negative examples, but abduction can be extended too [Inoue & Sakama, IJCAI-95].
- Theoretical results for one can be easily transferred to the other. E.g., The notion of **equivalence** is explored for abduction [Inoue & Sakama, MBR'04; IJCAI-05] and for induction [Sakama & Inoue, ILP'05].
- Computation can also be unified.

Inverse Entailment

Given that

$$B \wedge H \models E,$$

computing a hypothesis H can be done by

$$B \wedge \neg E \models \neg H.$$

I.e., $\neg H$ deductively follows from $B \wedge \neg E$.

Inverse Entailment

$$\textbf{\textit{B}}: \quad Human(Socrates),$$

$$\textbf{\textit{E}}: \quad Mortal(Socrates),$$

$$\textbf{\textit{H}}: \quad \forall x\ (\ Human(x) \supset Mortal(x)\ )$$

satisfies that:

$$B \wedge H \models E\ .$$

In fact,

$$B \wedge \neg E = Human(Socrates) \wedge \neg Mortal(Socrates)$$

$$\models \exists x\ (\ Human(x) \wedge \neg Mortal(x)\ ) = \neg H\ .$$

IE for **Abduction** [Inoue, 1992]

$$B \wedge \neg E \models \neg H$$

- Computation through *consequence finding*
- E : conjunction of (existentially-quantified) **literals**
- H : conjunctions of **literals**
- B : (full) clausal theory (**non-Horn** clauses)
- Note: Both $\neg E$ and $\neg H$ are **clauses**.
- sound and complete

IE for ILP [Muggleton, 1995]

$$B \land \neg E \models \neg H$$

- Use *consequence-finding* procedures twice
  [Yamamoto 1997]

- *B* : **Horn** clausal theory
- *E* : single **Horn clause**
- *H* : single (non-)Horn **clause**

- Note: Neither $\neg E$ nor $\neg H$ is a single clause, and both contain **existentially quantified variables**.

IE with $\bot$-clause: Incompleteness

Approach: Compute the ***$\bot$-clause***:

$$\bot(B, E) = \{ \neg L \mid L \text{ is a literal s.t. } B \wedge \neg E \models L \}.$$

Hypothesis H is constructed by generalizing $\bot$-clause:

$$H \models \bot(B, E).$$

- Sound but incomplete for recursive clauses [Yamamoto, 1997]
- Sufficient conditions for completeness [Furukawa et al., 1997; Yamamoto, 1997;1999]
- Incompleteness due to single-clause hypotheses [Ray, 2003]

Complete Calculus for IE

$B \wedge \neg E \models \neg H$

CF-Induction [Inoue, 2001]

- Compute the ***characteristic clauses*** of $B \wedge \neg E$
- Use any *consequence-finding* procedure.
- Use any *generalizer*.
- Includes the bottom method and abductive computation.

- *B* : full clausal theory (non-Horn clauses)
- *E* : full clausal theory (non-Horn clauses)
- *H* : full clausal theory (non-Horn clauses)

- Sound and complete

CF-Induction: Principle

$B \land H \models E$

$\Leftrightarrow B \land \neg E \models \neg H$

$\Leftrightarrow B \land \neg E \models Carc(B \land \neg E, \mathbf{P}) \models CC(B,E) \models \neg H$

$\Leftrightarrow CC(B,E) \subseteq Carc(B \land \neg E, \mathbf{P})$,

$\neg CC(B,E) \equiv F, \quad H \models F$ (where F is CNF)

CF-Induction: Algorithm

1. Compute $Carc(B \wedge \neg E, \mathbf{P})$.
2. Construct $CC(B,E)$ such that
   - $CC(B,E) \subseteq Carc(B \wedge \neg E, \mathbf{P})$;
   - $CC(B,E) \cap NewCarc(B, \neg E, \mathbf{P}) \neq \phi$.
3. Convert $\neg CC(B,E)$ into CNF F .
4. Generalize F to H such that
   - $B \wedge H$ is consistent.

CF-Induction: Generalizers

Given a CNF formula *F*, find a CNF formula *H* such that

$$H \models F.$$

- inverse Skolemization
- anti-instantiation
- anti-subsumption (dropping literals from clauses)
- anti-weakening (addition of clauses)
- inverse resolution
- Plotkin's least generalization

CF-Induction: Buntine's Example

B: $cat(x) \supset pet(x)$,

$small(x) \wedge fluffy(x) \wedge pet(x) \supset cuddly\_pet(x)$.

E: $fluffy(x) \wedge cat(x) \supset cuddly\_pet(x)$.

NewCarc($B, \neg E$, ***P***):

$fluffy(s_x)$, $cat(s_x)$, $\neg cuddly\_pet(s_x)$,

$pet(s_x)$, $\neg small(s_x)$

$CC(B,E) = NewCarc(B, \neg E, \mathbf{P})$

H: $fluffy(x) \wedge cat(x) \wedge pet(x) \supset cuddly\_pet(x) \vee small(x)$

CF-Induction: Yamamoto's Example

B: *even(0),*

odd(x) $\supset$ *even(s(x)).*

E: *odd(s(s(s(0)))).*

NewCarc(B, $\neg$*E,* ***P****):* $\neg$*odd(s(s(s(0)))).*

CC(B,E) : *even(0), odd(s(0))* $\supset$ *even(s(s(0))),* $\neg$*odd(s(s(s(0)))).*

CNF($\neg$*CC(B,E)* **):**

even(0) $\supset$ *odd(s(0))* $\vee$ *odd(s(s(s(0)))),*

even(0) $\wedge$ *even(s(s(0)))* $\supset$ *odd(s(s(s(0)))).*

H: *even(x)* $\supset$ *odd(s(x)).*

Induction v.s. Abduction

- CF-induction is realized by abductive proc.
- CF-induction includes abduction.
- Abduction comprises of computing *NewCarc(B,* $\neg$*E,* ***P****)* only.
- Induction often requires formulas in *Carc(B* $\wedge$ $\neg$*E,* ***P****)* $-$ *NewCarc(B,* $\neg$*E,* ***P****)*. Namely, <u>background knowledge</u> is associated with observations.

Yamamoto & Fronhőfer's Example

B: $dog(x) \wedge small(x) \supset pet(x)$.
E: $pet(c)$.

$NewCarc(B, \neg E, \mathbf{P})$: $\neg pet(c)$, $\neg dog(c) \vee \neg small(c)$.

$CC(B,E) = NewCarc(B, \neg E, \mathbf{P})$.

$\neg CC(B,E)$ **:** $pet(c) \vee (dog(c) \wedge small(c))$.

CNF($\neg CC(B,E)$ **):** $pet(c) \vee dog(c)$, $pet(c) \vee small(c)$.

H: $pet(x) \vee dog(x)$, $pet(x) \vee small(x)$.

Muggleton's Example

B: *white(swan1).*

E: $\neg$*black(swan1).*

NewCarc(B, $\neg$*E,* ***P****):* *black(swan1).*

CC(B,E) : *white(swan1), black(swan1).*

$\neg$*CC(B,E)* **:** $\neg$*white(swan1)* $\vee$ $\neg$*black(swan1).*

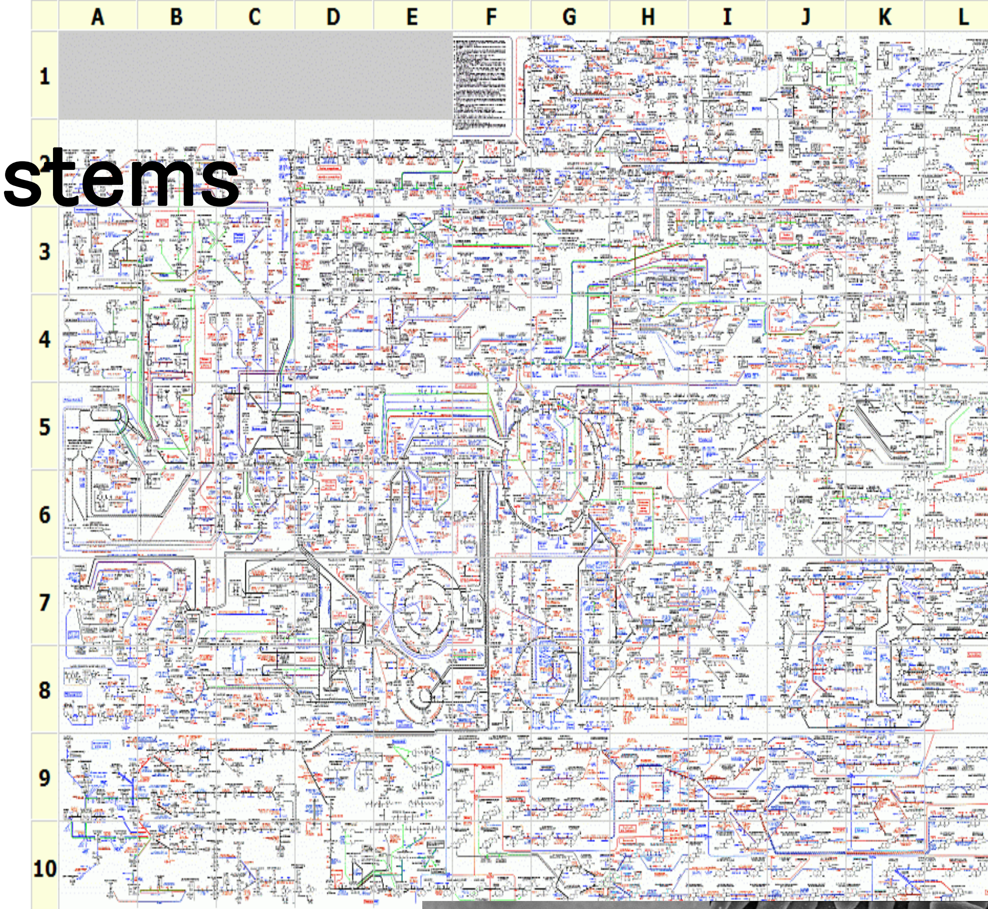
H: $\neg$*white(x)* $\vee$ $\neg$*black(x).*

Automated biological discovery

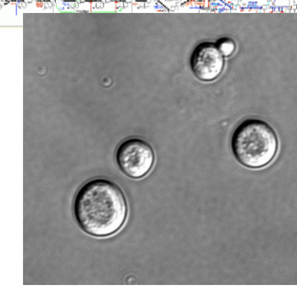
- **“Robot Scientist”**
  - King, R.D. *et al.*, “Functional Genomic Hypothesis Generation and Experimentation by a Robot Scientist”, *Nature*, 427, 2004.
  - King, R.D. *et al.*, “The Automation of Science”, *Science*, 324, 2009.
- **“Chemical Turing Machine”**
  - Muggleton, S.D., “Exceeding Human Limits”. *Nature*, 440, 2006.

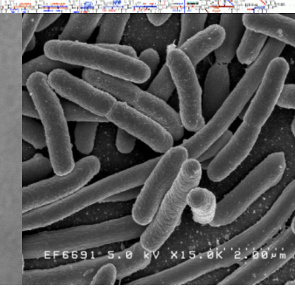
CHANGES TO TRADITIONAL SCIENCE WITH AUTOMATION	
Traditional science	**Automated science**
Hypotheses	Machine-encoded logical hypotheses
Chemical knowledge	Machine-encoded chemical algebra
Experiments	Chemical Turing machine programs
Experimental design	Decision theory

Modeling Biological Systems



- Explain and predict metabolic pathways.
  - Generic Model:
    - *Saccharomyces Cerevisiae*
    - *E-coli*
  - Biological Phenomenon can be explained by *Inductive Logic Programming* (**ILP**).



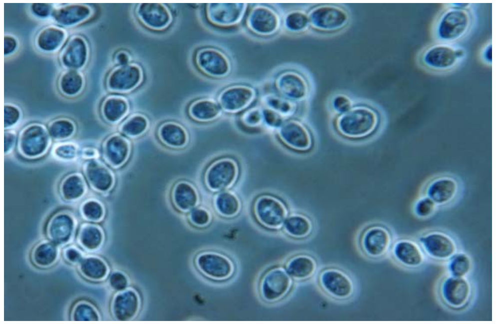


Inductive Learning Approaches

- Goals
  - Finding inhibitions in a metabolic pathway.
  - Discovering causal rules which augment an incomplete background theory.
  - Predicting changes of concentration in intracellular fluxes.
- Previous Work
  - Using an abductive logic programming technique on the problem of inhibitions of metabolic pathways at steady states (Tamaddoni-Nezdah et al., 2006)
- New Approach (Yamamoto, Inoue & Doncescu, 2007)
  - Integration of abduction and induction.
  - Not only steady states but also dynamic models.

Prediction of Intracellular Fluxes

- **Goals**
  - Predicting concentration changes in intracellular metabolites
  - Discovering causal laws augmenting an incomplete background theory



- **Approaches**

  Inverse Entailment for induction (CF-induction)

  - Examples E :

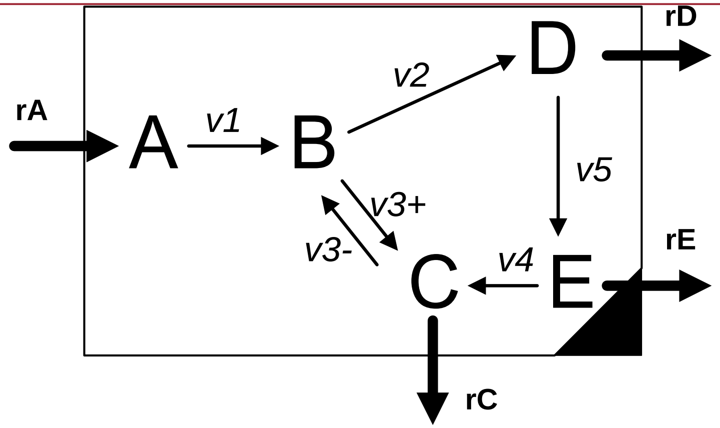
    changes (up/down) of concentrations of extracelluar metabolites

  - Background theory B :
    - chemical reactions in a metabolic networks
    - clauses concerning known inhibitory effects

  - Hypothesis H :
    - a clausal theory which consists of both literals whose predicate is "inhibition" and clauses corresponding to causal laws

Metabolite Balancing

- **Intracellular fluxes** are determined as a function of the measurable extracellular fluxes using a stoichiometric model for major intracellular reactions and applying a mass balance around each intracellular metabolite.



$$\begin{cases} v1 = -rA \\ v2 = v1 - a \\ v4 = rC - a \\ v5 = v1 - rD - a \\ v5 = rE + rC - a \\ a = [(v3+) - (v3-)] \end{cases}$$

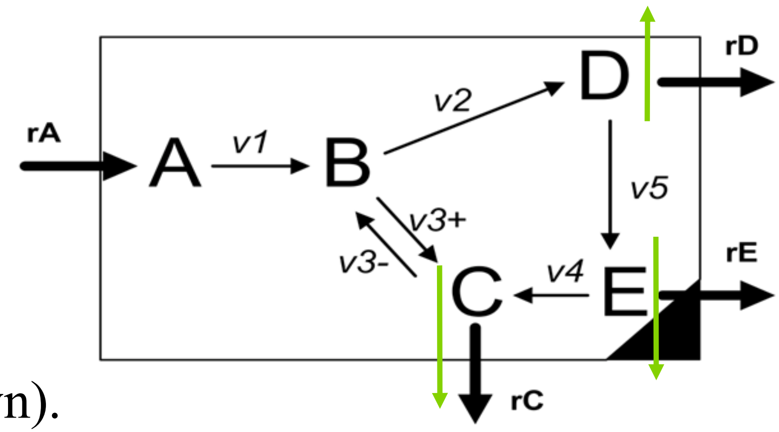
v1, v2, v3+, v3-, v4 : unknown fluxes at the steady stat

rA, rC, rD, rE : metabolite extracellular accumulation rat

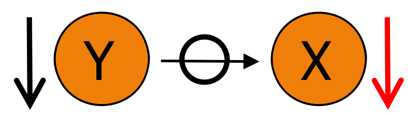
Example: Unobservable Metabolite

- B :

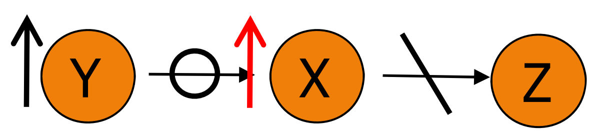
concentration(a, up).
reaction(a, b).   reaction(b, d).
reaction(d, e).   reaction(e, c).
reaction(c, b).   reaction(b, c).
$\neg$concentration(X, up) $\leftarrow$ concentration(X, down).



concentration(X, down) $\leftarrow$
reaction(Y, X), $\neg$inhibited(Y, X), concentration(Y, down).



concentration(X, up) $\leftarrow$ concentration(Y, up), reaction(Y, X), reaction(X, Z),
$\neg$inhibited(Y, X), inhibited(X, Z).



- E :

concentration(d, up).   concentration(e, down).   concentration(c, down).

Example: Outputs of CF-induction

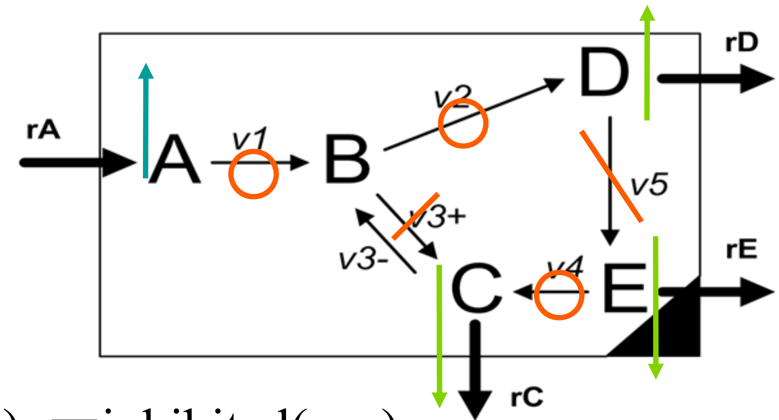
- H_1 :

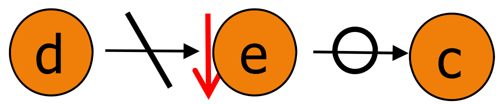
¬inhibited(a, b). inhibited(b, c).

¬inhibited(e, c). inhibited(d, e).

¬inhibited(b, d).

concentration(e, down) ← inhibited(d, e), ¬inhibited(e, c).



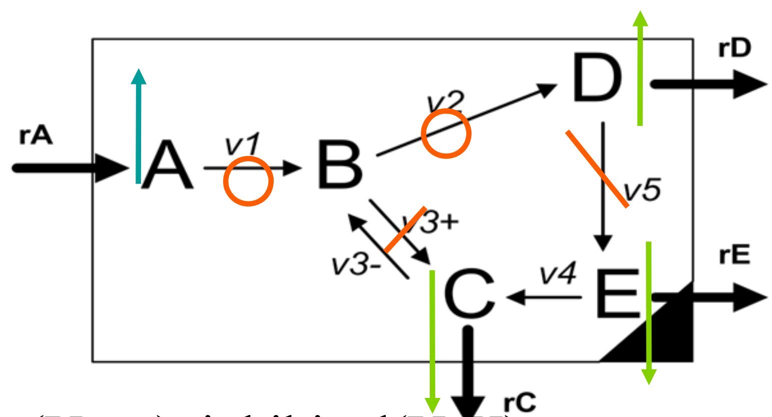


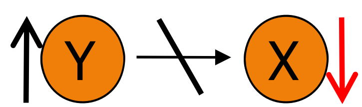
- H_2 :

¬inhibited(a, b). inhibited(b, c).

¬inhibited(b, d). inhibited(d, e).

concentration(X, down) ← concentration(Y, up), inhibited(Y, X).





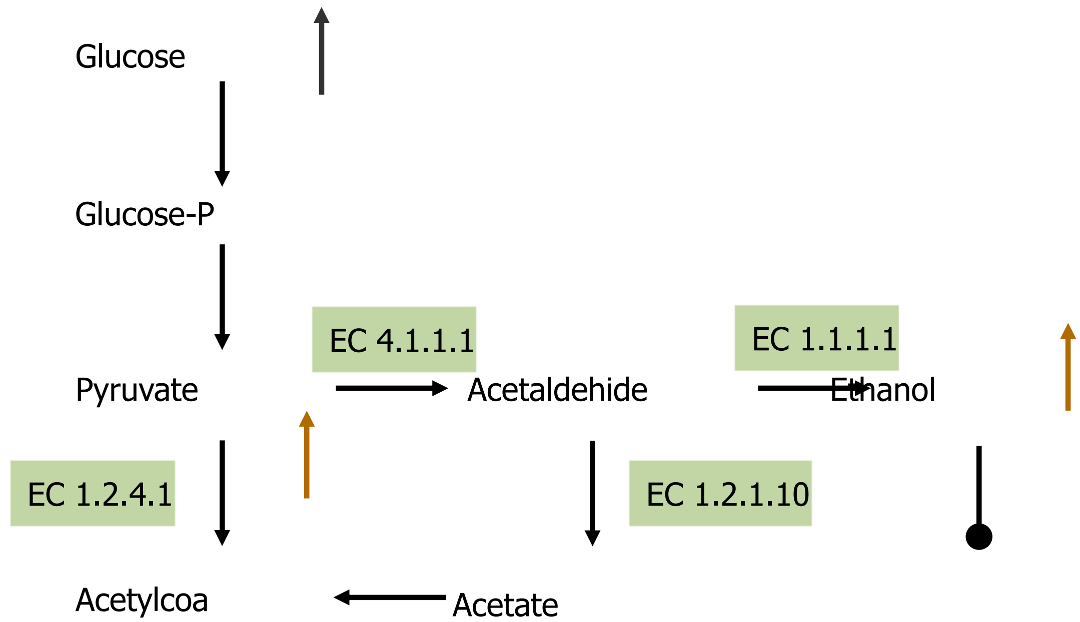
Example: Metabolic Pathway (Pyruvate)

- ***B***:

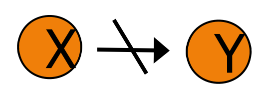
reaction(pyruvate, acetylcoa).
reaction(pyruvate, acetaldehide).
reaction(glucose, glucosep).
reaction(glucosep, pyruvate).
reaction(acetaldehide, acetate).
reaction(acetate, acetylcoa).
reaction(acetaldehide, ethanol).

concentration(glucose, up).

terminal(ethanol).



blocked(X) $\leftarrow$ reaction(X,Y), inhibited(X,Y).



blocked(X) $\leftarrow$ terminal(X).

concentration(X,up) $\leftarrow$ reaction(Y,X), $\neg$inhibited(Y,X), blocked(X).

E : concentration(ethanol,up). concentration(pyruvate, up).

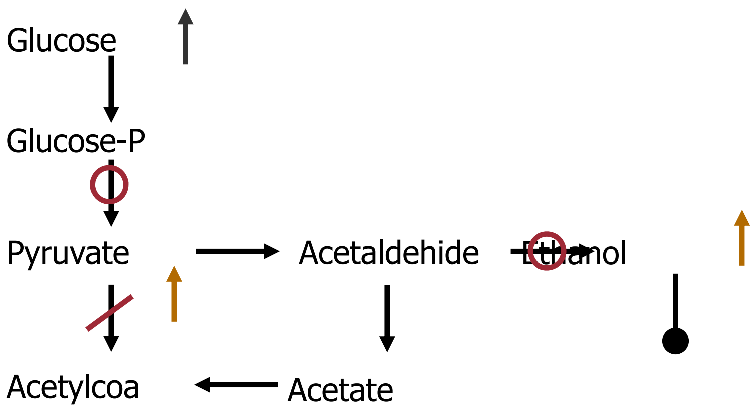
Example: Outputs of CF-induction

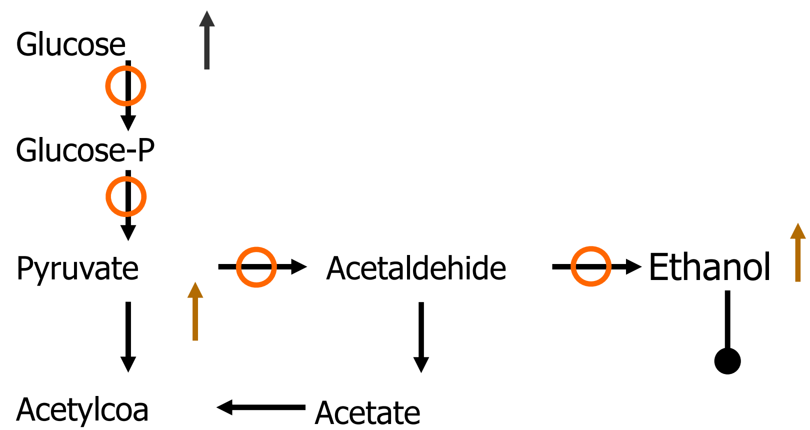
- ***H*₁:**

¬Inhibited(glucosep, pyruvate).
¬inhibited(acetaldehide, ethanol).
inhibited(pyruvate, acetylcoa).

- ***H*₂:**

¬inhibited(glucose, glucosep)
¬Inhibited(glucosep, pyruvate).
¬inhibited(acetaldehide, ethanol).
¬inhibited(pyruvate, acetaldehide).
concentration(Y, up) ←
¬inhibited(X, Y), concentration(X, up).





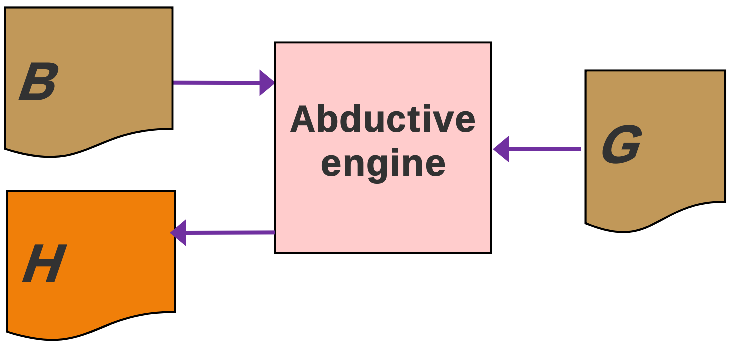
Abduction: Logical Framework

Input:

- ***B*** : background theory
- ***G*** : observations
- ***Γ*** : possible causes (abducibles)

Output:

- ***H*** : hypothesis satisfying that
  - $B \wedge H \models G$
  - $B \wedge H$ is consistent
  - ***H*** is a set of instances of literals from ***Γ***.



Inverse Entailment (IE)

Computing a hypothesis *H* can be done **deductively** by:

$$B \wedge \neg G \models \neg H$$

*We use a **consequence finding** technique for IE computation.*

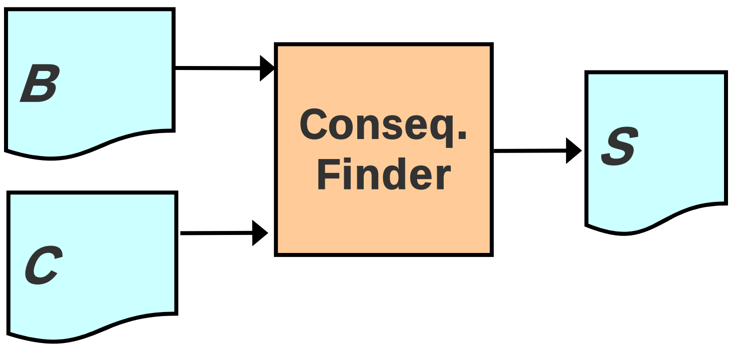
Consequence Finding

Input:

- ***B*** : first-order (clausal) theory
- ***C*** : "new" clausal theory
- ***P*** : language restriction (*"production field"*)

Output:

- ***S*** : the (subsumption-minimal) "new" consequences satisfying that
  - $B \wedge C \models S$
  - $B \not\models S$
  - ***S* belongs to *P*.**



- **SOL-resolution** (Inoue, IJCAI-91)
- **SOLAR** (Nabeshima, Iwanuma & Inoue, TABLEAUX'03)

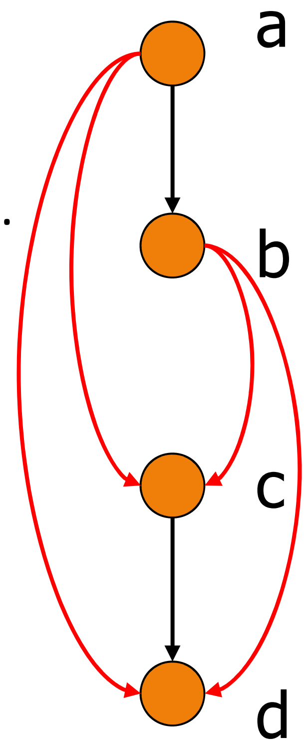
■ For Theorem Proving, ***C*** is the negation of the target theorem and ***S*** is the empty clause (generalization of *proof-finding*).

■ For Inverse Entailment, $C = \neg G$, $S = \neg H$, and $P = \neg \Gamma$.

Inverse Entailment for Abduction

SOLAR Example: **graph completion problem** – *pathway finding*

Find an arc which enables a path from A to D.



Background theory:

path(X,Y) ← node(X) ∧ node(Y) ∧ arc(X,Y).

path(X,Z) ← node(X) ∧ node(Y) ∧ node(Z) ∧ arc(X,Y) ∧ path(Y,Z).

node(a). node(b). node(c). node(d). arc(a,b). arc(c,d).

Negated observation:

¬path(a,d).

Production field:

literal form = [¬arc(_,_)] & clause length ≤ 1.

Output of SOLAR:

1. ¬arc(a, d). 2. ¬arc(a, c). 3. ¬arc(b, c). 4. ¬arc(b, d).

Abductive Inference (Naïve Formalization)

given

B	theory (set of clauses)
E	goal (set of literals)
A	abducibles (set of possible assumptions)

find

$H \subseteq A$	explanation (set of assumptions)
σ	answer (variable bindings)

where

$$B \wedge H \models E\sigma$$

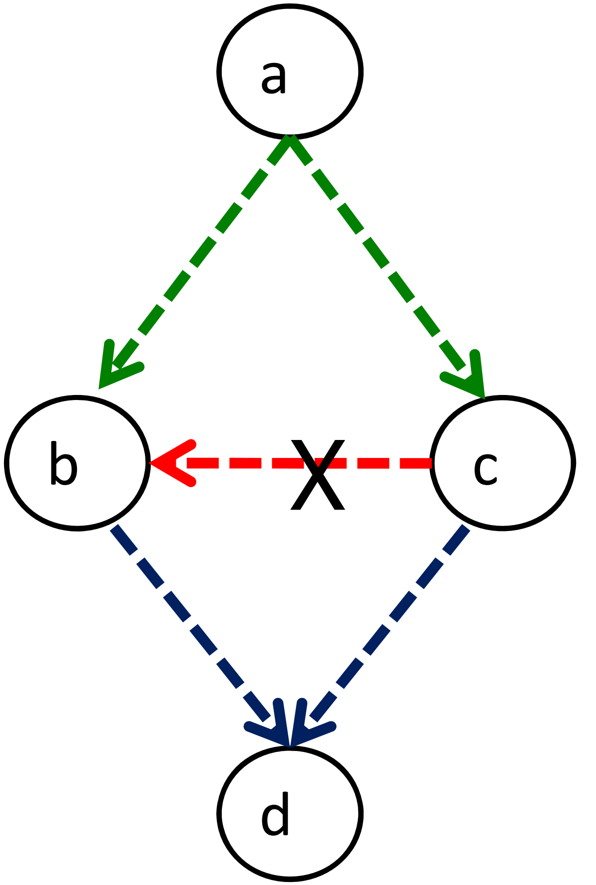
Implicitly: B is a universally quantified conjunction and
E is an existentially quantified conjunction.

Metabolic Pathway (Ray & Inoue, DS'07)

$$B = \begin{cases} pathway(X,Z) \leftarrow reaction(X,Y) \wedge pathway(Y,Z) \\ pathway(X,Z) \leftarrow reaction(X,Z) \\ reaction(a,b) \vee reaction(a,c) \\ reaction(b,d) \vee reaction(c,d) \\ \neg reaction(c,b) \end{cases}$$

Metabolic Pathway (Ray & Inoue, DS'07)

$$B = \begin{cases} pathway(X,Z) \leftarrow reaction(X,Y) \wedge pathway(Y,Z) \\ pathway(X,Z) \leftarrow reaction(X,Z) \\ reaction(a,b) \vee reaction(a,c) \\ reaction(b,d) \vee reaction(c,d) \\ \neg reaction(c,b) \end{cases}$$

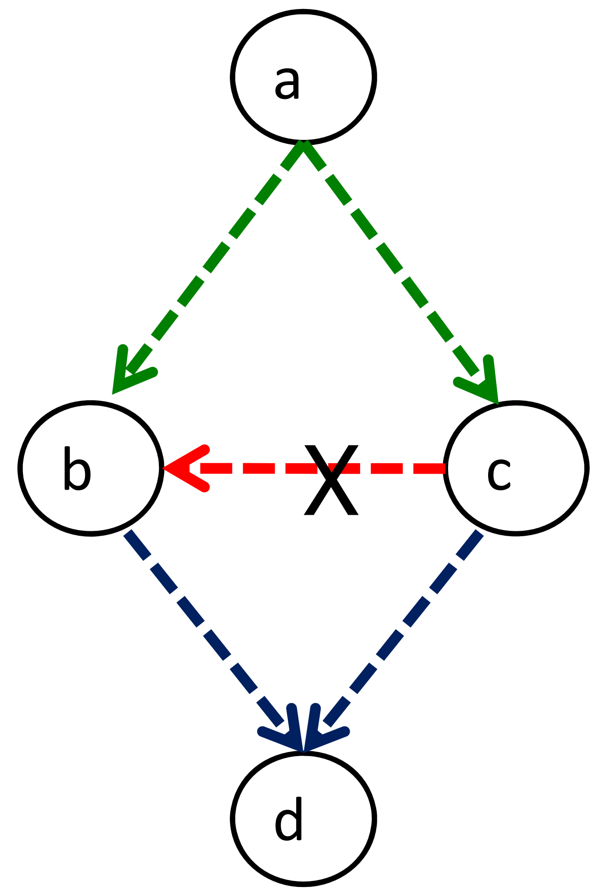


Metabolic Pathway (Ray & Inoue, DS'07)

$$B = \begin{cases} pathway(X,Z) \leftarrow reaction(X,Y) \wedge pathway(Y,Z) \\ pathway(X,Z) \leftarrow reaction(X,Z) \\ reaction(a,b) \vee reaction(a,c) \\ reaction(b,d) \vee reaction(c,d) \\ \neg reaction(c,b) \end{cases}$$

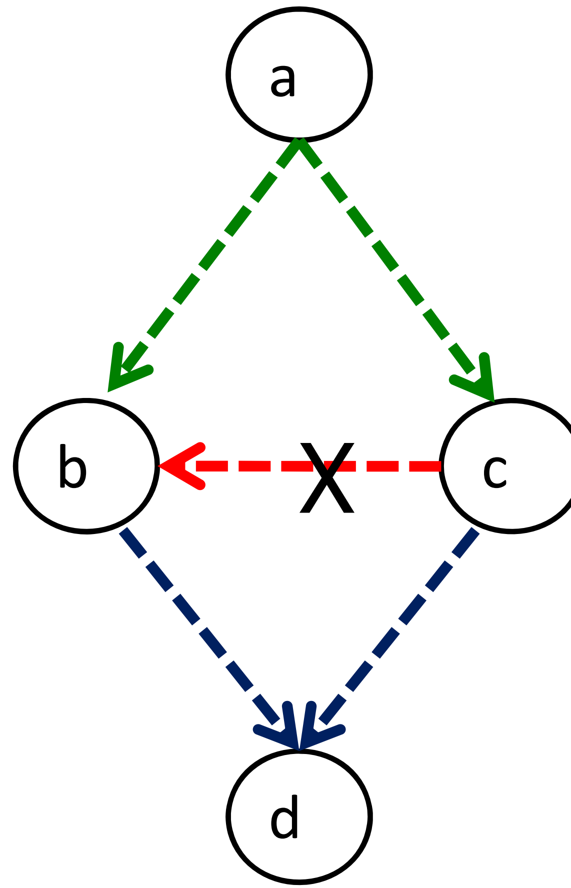
$$E = \{pathway(U,d)\}$$

$$A = \{reaction(V,W)\}$$



Metabolic Pathway (Ray & Inoue, DS'07)

$$T = \begin{cases} pathway(X,Z) \leftarrow reaction(X,Y) \wedge pathway(Y,Z) \\ pathway(X,Z) \leftarrow reaction(X,Z) \\ reaction(a,b) \vee reaction(a,c) \\ reaction(b,d) \vee reaction(c,d) \\ \neg reaction(c,b) \end{cases}$$



$$E = \{pathway(U,d)\}$$

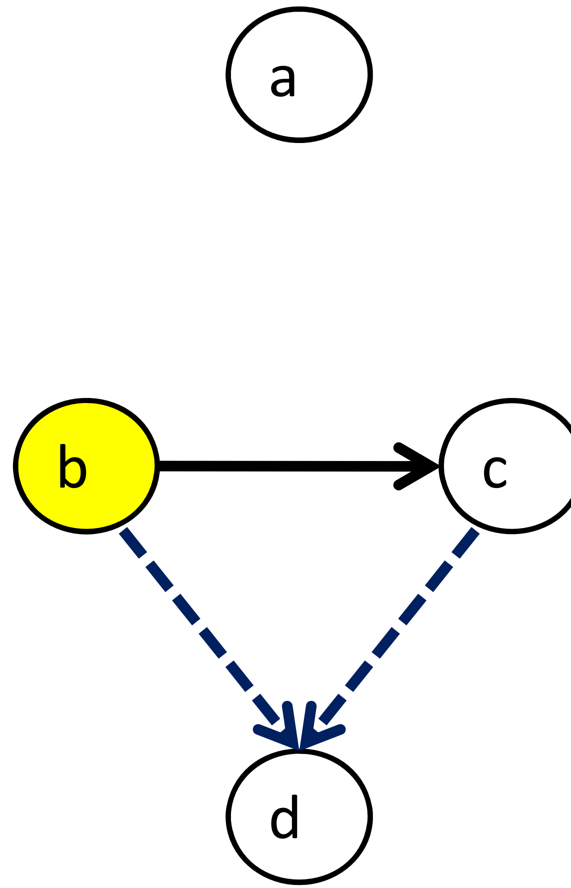
% (from which U) is there a path to d?

$$A = \{reaction(V,W)\}$$

% assuming reactions from some V to W

Metabolic Pathway: A Solution

$$B = \begin{cases} pathway(X,Z) \leftarrow reaction(X,Y) \wedge pathway(Y,Z) \\ pathway(X,Z) \leftarrow reaction(X,Z) \\ reaction(a,b) \vee reaction(a,c) \\ reaction(b,d) \vee reaction(c,d) \\ \neg reaction(c,b) \end{cases}$$



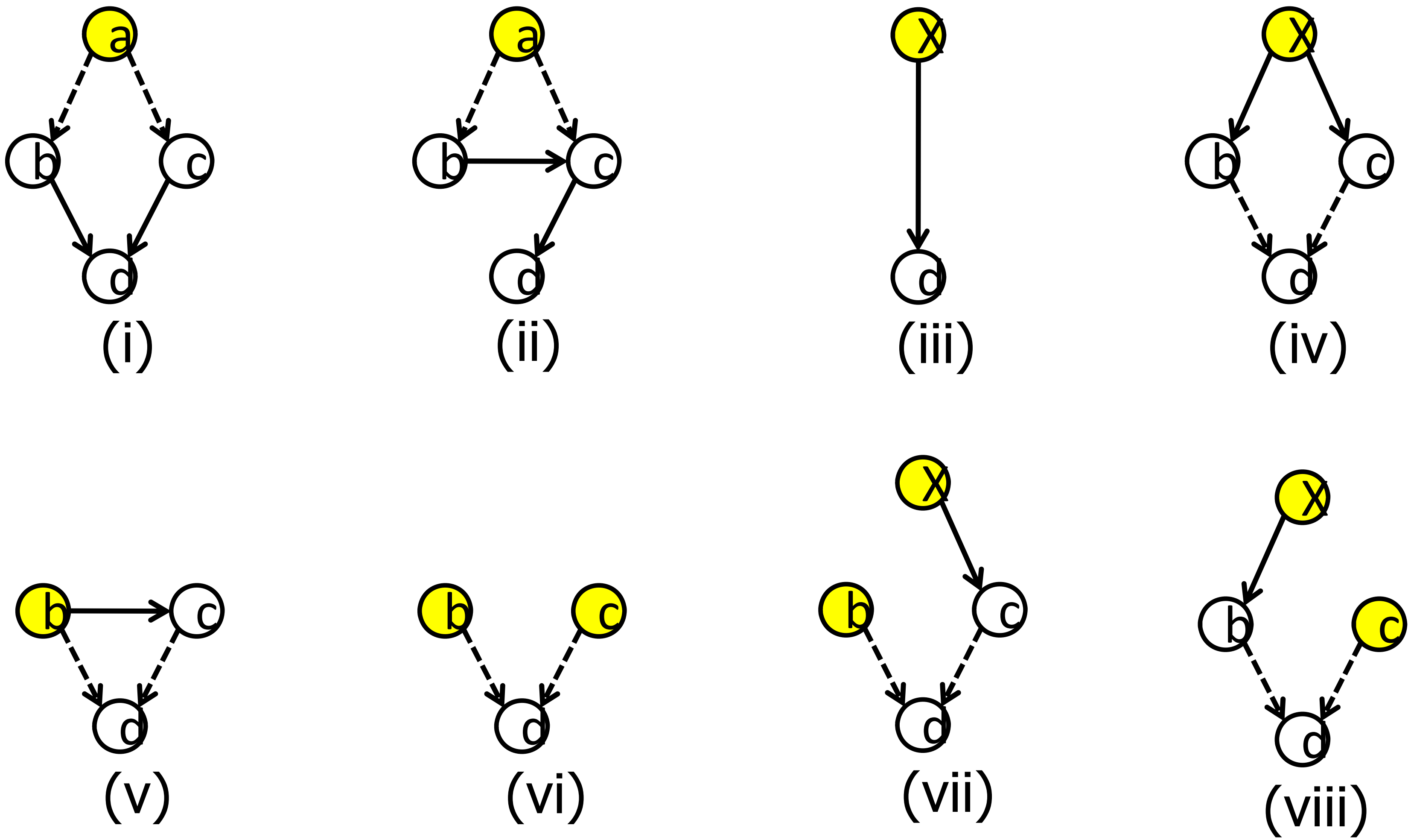
$$E\sigma = \{pathway(b,d)\}$$

% there is a path from b to d

$$H = \{reaction(b,c)\}$$

% assuming a reaction from b to c

Metabolic Pathway: MORE Solutions



Problem: want to express <u>non-ground answers</u> like (iii), and <u>disjunctive answers</u> such as (vi).

Abductive Inference (Revisited)

given

B	theory (set of clauses)
E	goal (set of literals)
A	abducibles (set of possible assumptions)

find

$H \subseteq A$	explanation (set of assumptions)
σ	answer (variable bindings)

where

$$B \wedge H \models E\sigma$$

However: No interaction between variables in H and E and no way to return disjunctive answers in σ.

Abductive Inference (Ray & Inoue, DS'07)

given

B	theory (background knowledge)
E	goal (set of given observations)
A	abducibles (set of possible assumptions)

find

$H \subseteq A$	explanation (set of assumptions)
Θ	answer (SET of variable bindings)

where

$$B \models \forall \left(\bigwedge_{L \in H} L \rightarrow \bigvee_{\sigma \in \Theta} E\sigma \right)$$

Abductive Inference (Ray & Inoue, DS'07)

given

B	theory (background knowledge)
E	goal (set of given observations)
A	abducibles (set of possible assumptions)

find

$H \subseteq A$	explanation (set of assumptions)
Θ	answer (SET of variable bindings)

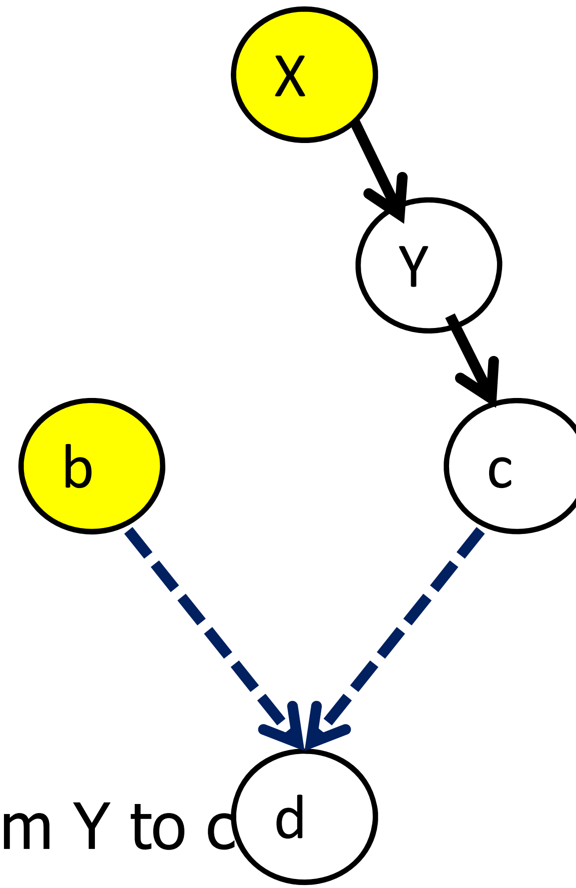
where

$$B \models \forall \left(\bigwedge_{L \in H} L \rightarrow \bigvee_{\sigma \in \Theta} E\sigma \right)$$

% the conjunction of assumptions H implies the disjunction of answers Θ.

Metabolic Pathway: Another Solution

$$B = \begin{cases} pathway(X,Z) \leftarrow reaction(X,Y) \wedge pathway(Y,Z) \\ pathway(X,Z) \leftarrow reaction(X,Z) \\ reaction(a,b) \vee reaction(a,c) \\ reaction(b,d) \vee reaction(c,d) \\ \neg reaction(c,b) \end{cases}$$



$$\Theta = \{\{U/b\},\{U/X\}\}$$

% there is a path from b or X to d

$$H = \{reaction(X,Y), reaction(Y,c)\}$$

% assuming reactions from X to Y and from Y to c

Evaluating Abductive Hypotheses using an EM Algorithms on BDDs

Katsumi Inoue[1,2], **Taisuke Sato**[2,1],
Masakazu Ishihata[2], **Yoshitaka Kameya**[2],
Hidetomo Nabeshima[3]

[1] National Institute of Informatics
[2] Tokyo Institute of Technology
[3] University of Yamanashi

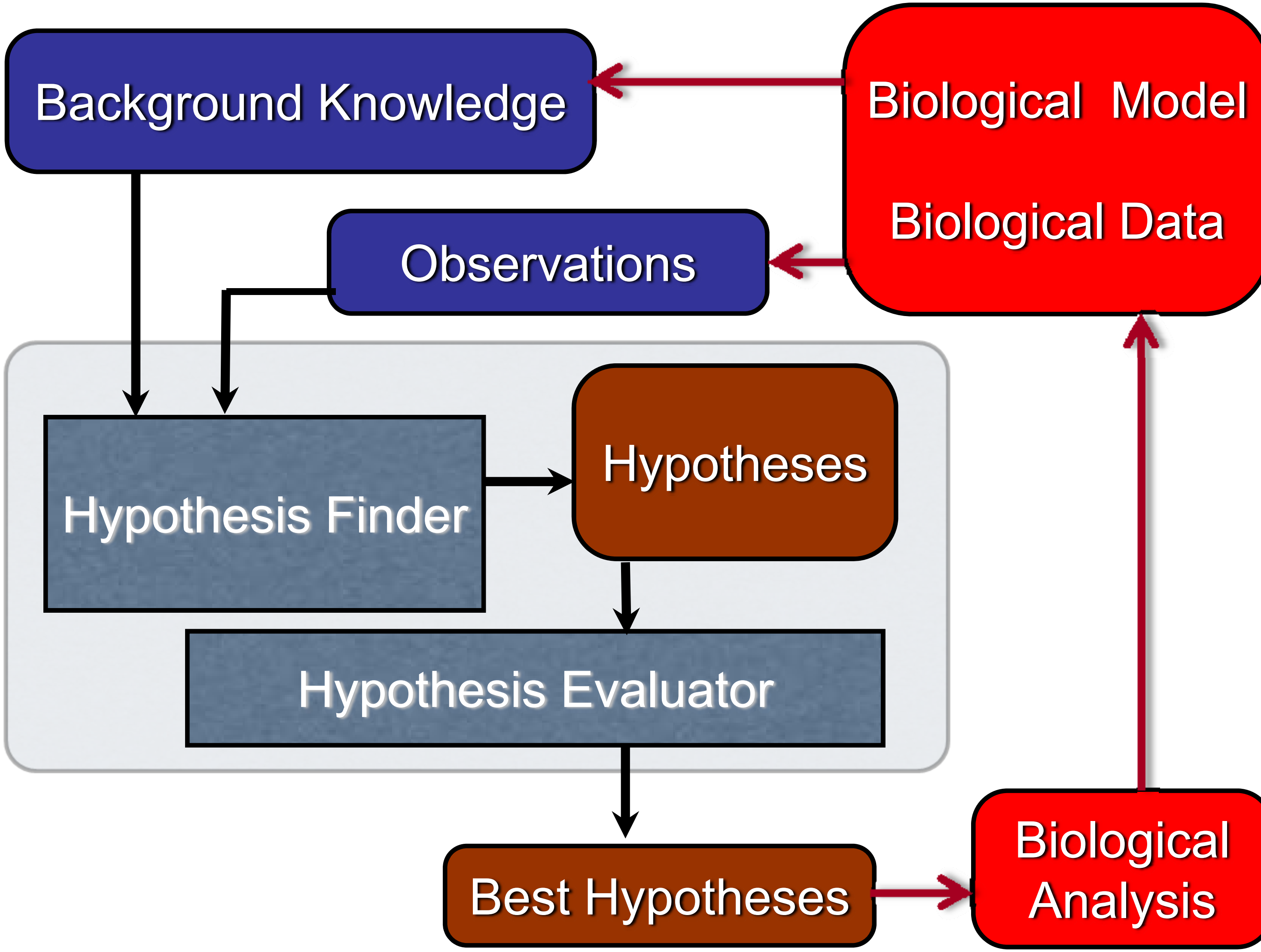
Thanks to: **Yoshitaka Yamamoto, Koji Iwanuma, Andrei Doncescu, Stephen Muggleton, Oliver Ray, Takehide Soh, JST** and **JSPS.**

adapted from presentation at IJCAI-09

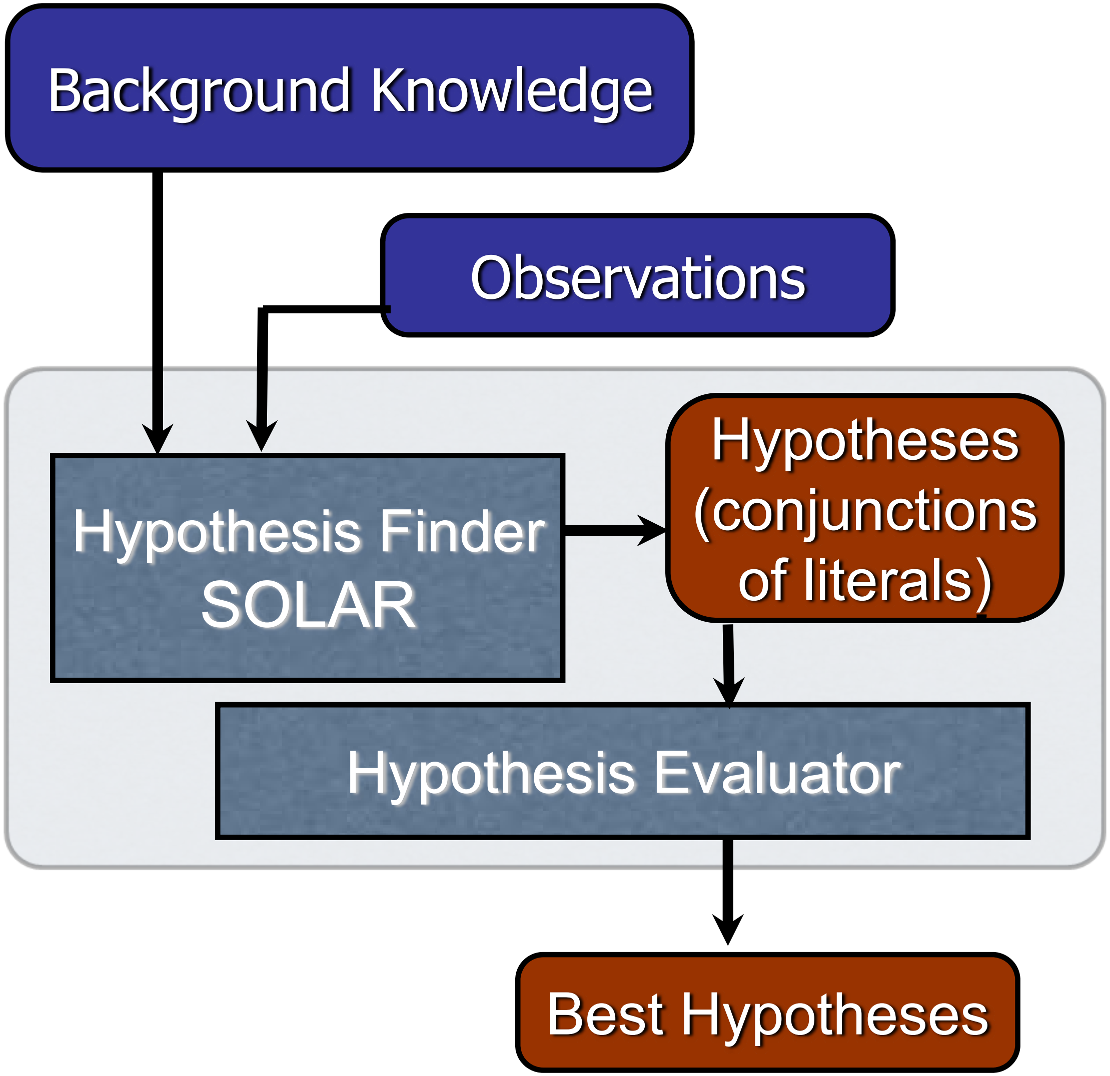
Abduction and discovery

- Application of abduction to **scientific discovery**
  - (Zupan *et al*., *Bioinformatics* 2003), (King *et al*., *Nature* 2004; *Science* 2009), (Muggleton, *Nature* 2006), etc.
- Knowledge is structured as a **network**
- Knowledge and data bases are **incomplete**
  - Constraints are often very weak, so there exist a large number of logically possible hypotheses
- Hypotheses are composed for **multiple observations**
  - 20 metabolites, 10 explanations for each – 10^{20}
- **Hypothesis evaluation** is indispensable, but how?

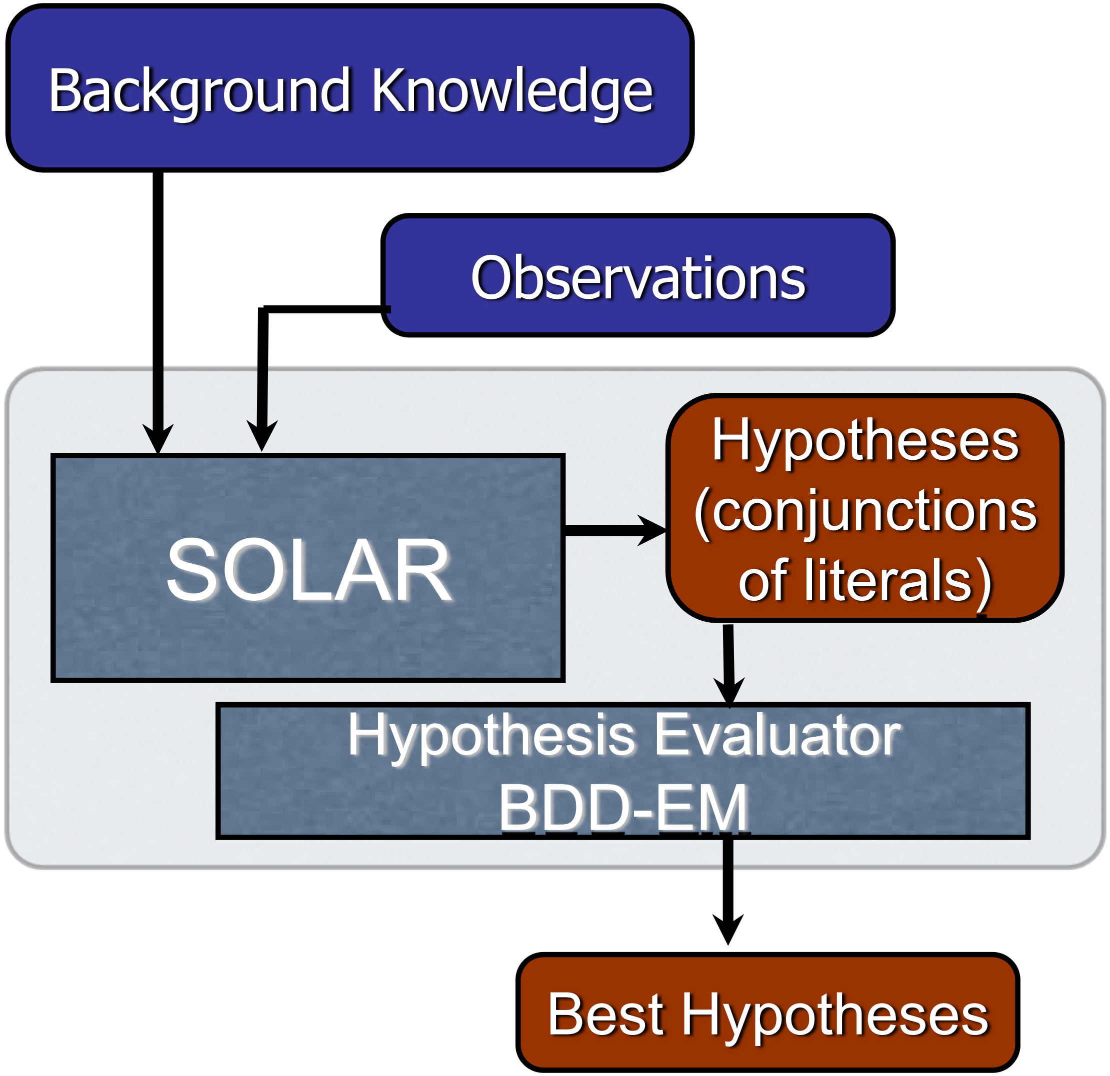
An abductive system architecture



The current abductive system



The current abductive system



BDD-EM algorithm

- Learning probabilities of a model described by a Boolean formula of propositions and their probabilities from the observations.

- **BDD-EM algorithm** (Ishihata *et al*., 2008)
  - **The EM algorithm**: maximum likelihood estimation
  - **Binary decision diagrams** (BDDs): compact expression of Boolean formulas.

BDD-EM algorithm = BDD + EM algorithm

Problem setting

Example: $F \Longleftrightarrow X1 \vee (X2 \wedge \neg X3)$

F : observable variable

Xi : basic variable (unobservable)

X1, ***X2*** $\wedge \neg$ ***X3*** : hypotheses (or constraints)

☺ We can observe a value f ($\in \{0,1\}$) of F,

☹ but cannot observe values of $X1$, $X2$ and $X3$.

😐 What we want to know is the most probable hypothesis that account for the observation F.

Probabilities of hypotheses

Probability of a hypothesis is computed as the product of probabilities of basic variables ***X1***, ***X2*** and ***X3***.

$H1 = X1$ $\quad P(H1) = \theta_{X1=1}$

$H2 = X2 \wedge \neg X3$ $\quad P(H2) = \theta_{X2=1}\,\theta_{X3=0}$

$\theta_{Xi=x} \equiv P(Xi=x)$

$x \in \{0,1\}$

Maximum likelihood estimation (MLE):

Find $\theta_{X1=x}$, $\theta_{X2=x}$ and $\theta_{X3=x}$

maximizing the likelihood $P(F=f)$ of the observation.

EM algorithm (Dempster *et al.*, 1977)

Iterative MLE method from incomplete data in which values of basic variables are unknown.

E-step: Compute conditional expectations

$$E[Xi=x \mid F=f\,] \;:=\; E[Xi=x, F=f] \,/\, P(F=f)$$

M-step: Update probabilities

$$\theta_{Xi=x} \;:=\; E[Xi=x \mid F=f] \,/\, \left(\textstyle\sum_y E[Xi=y \mid F=f]\right)$$

Iterate E- & M-steps until the likelihood saturates.

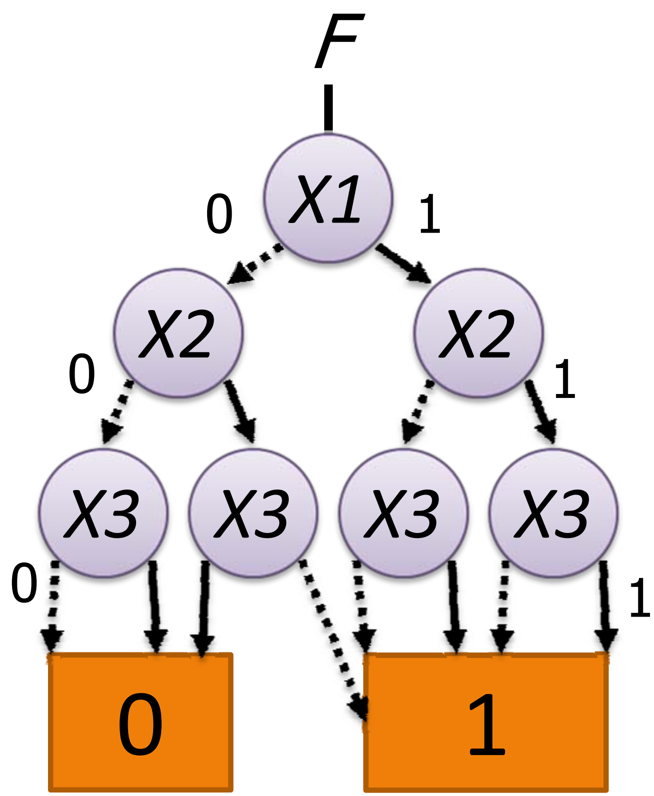
Binary decision diagrams (BDDs)

compressed expressions of Boolean formulas.

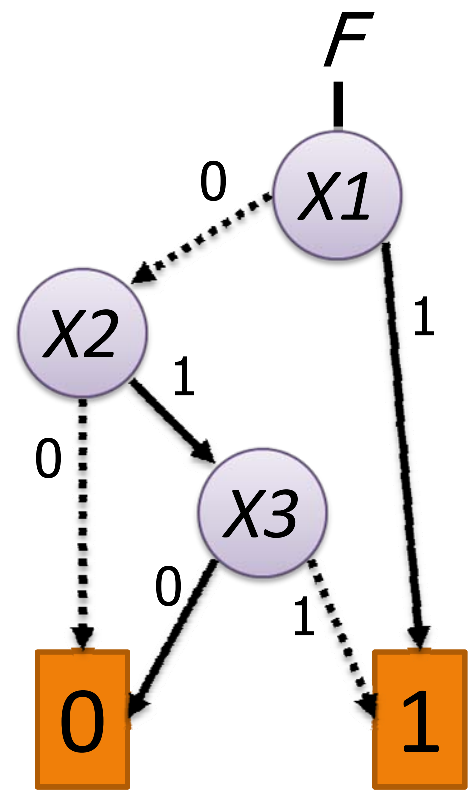
$F \Leftrightarrow X1 \vee (X2 \wedge \neg X3)$

X1	*X2*	*X3*	*F*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Truth table



Binary decision tree



BDD

Probability computation on BDDs

On the truth table for *F*

Sum of prob. of rows representing *F*=1 in the truth table for *F*.

X1	*X2*	*X3*	*F*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

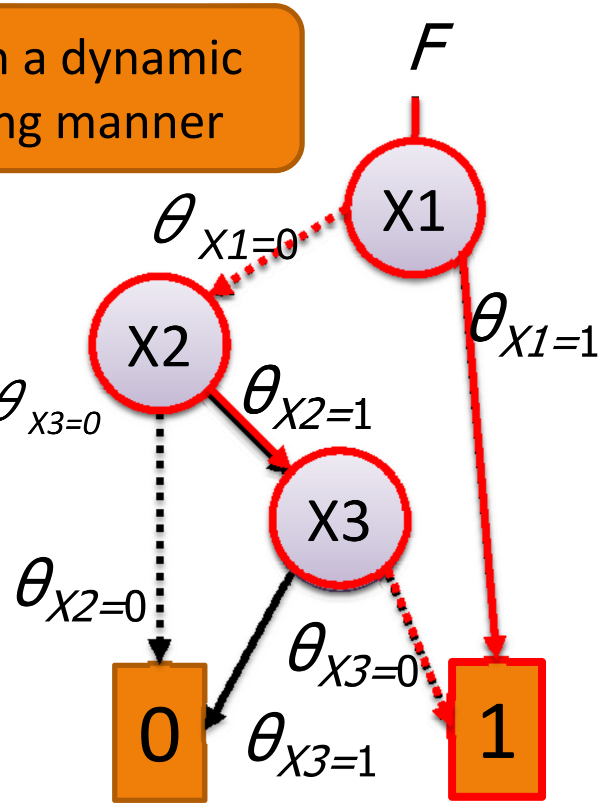
$$P(F=1) = \theta_{X1=0}\,\theta_{X2=1}\,\theta_{X3=0} + \theta_{X1=1}\,\theta_{X2=0}\,\theta_{X3=0} + \theta_{X1=1}\,\theta_{X2=0}\,\theta_{X3=1} + \theta_{X1=1}\,\theta_{X2=1}\,\theta_{X3=0} + \theta_{X1=1}\,\theta_{X2=1}\,\theta_{X3=1}$$

On the BDD for *F*

Sum of prob. of paths representing *F*=1 in the truth table for *F*.

Computed in a dynamic programming manner

$$P(F=1) = \theta_{X1=1} + \theta_{X1=0}\,\theta_{X1=1}\,\theta_{X3=0}$$

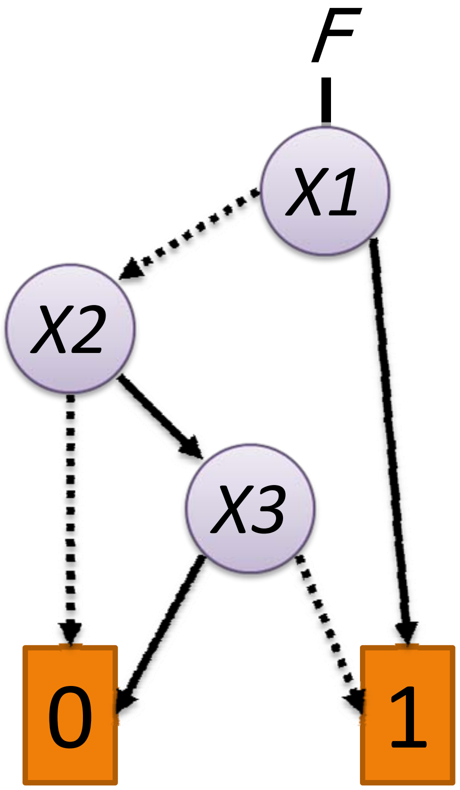


Expectation computation on BDDs

Computation of expectations *E*[*Xi=1,F=f*] is carried out by *forward* and *backward* probabilities for each nodes.

1. Forward probabilities $F_{\theta}[Xi]$
   Sum of probabilities of paths from the root node to *Xi*.

2. Backward probabilities $B_{\theta}[Xi]$
   Sum of probabilities of paths from *Xi* to the terminal 1.



Forward and backward probabilities

Forward probabilities

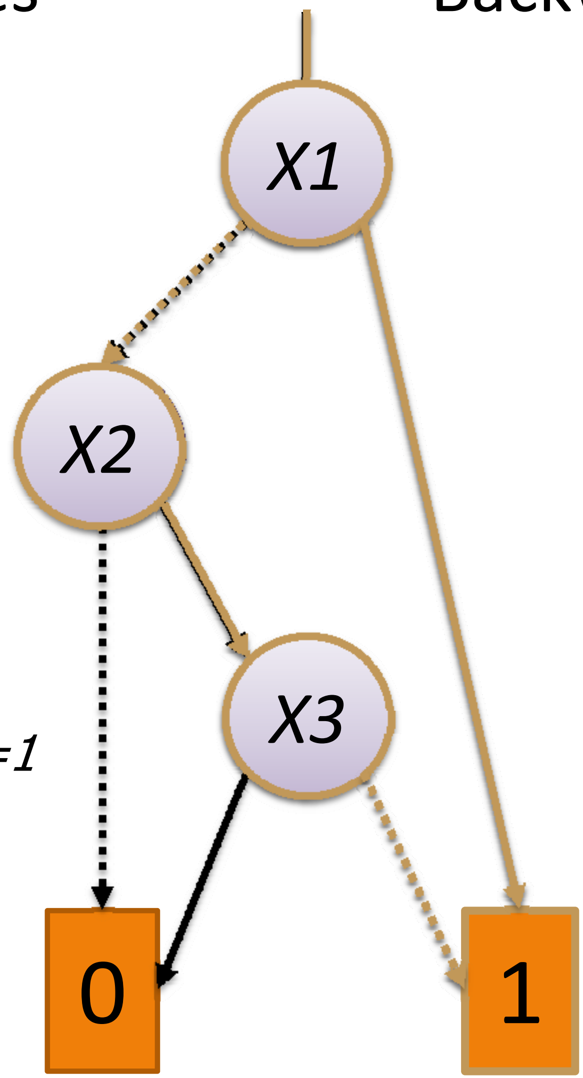
Backward probabilities

$F_\theta[X1] = 1$

$F_\theta[X2] = \theta_{X1=0}$

$F_\theta[X3] = \theta_{X1=0}\theta_{X2=1}$

$F_\theta[1] = \theta_{X1=1} + \theta_{X1=0}\theta_{X2=1}\theta_{X3=0}$



Forward and backward probabilities

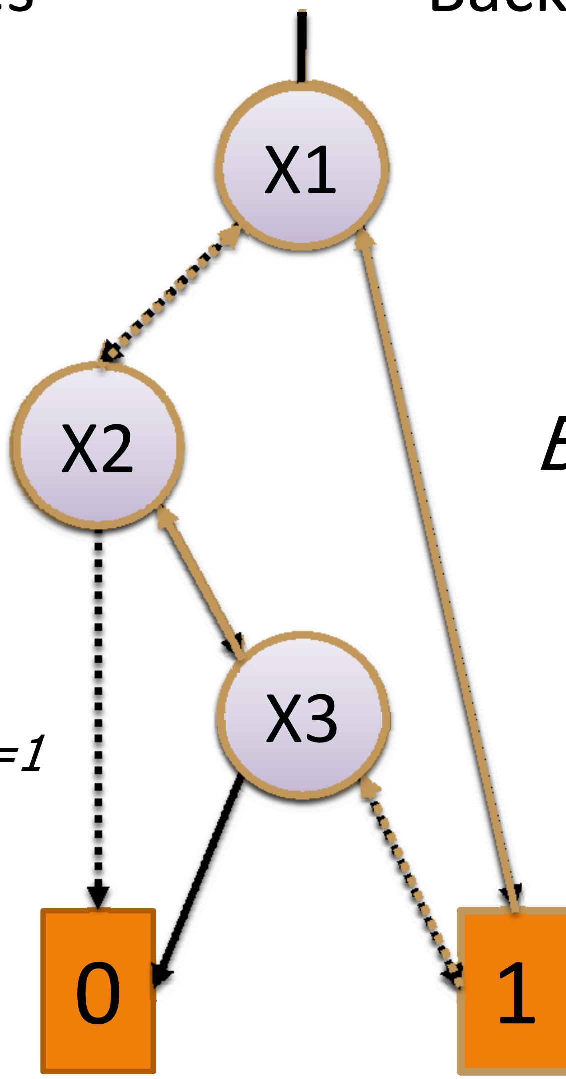
Forward probabilities

$F_\theta[X1] = 1$

$F_\theta[X2] = \theta_{X1=0}$

$F_\theta[X3] = \theta_{X1=0}\theta_{X2=1}$

$F_\theta[1] = \theta_{X1=1} + \theta_{X1=0}\theta_{X2=1}\theta_{X3=0}$



Backward probabilities

$B_\theta[X1] = \theta_{X1=1} + \theta_{X3=0}\theta_{X2=1}\theta_{X1=0}$

$B_\theta[X2] = \theta_{X3=0}\theta_{X2=1}$

$B_\theta[X3] = \theta_{X3=0}$

$B_\theta[1] = 1$

BDD-EM algorithm

E-step : Compute conditional expectations

$$E[Xi{=}x \mid F{=}f\,] := E[Xi{=}x, F{=}f] \,/\, P(F{=}f)$$

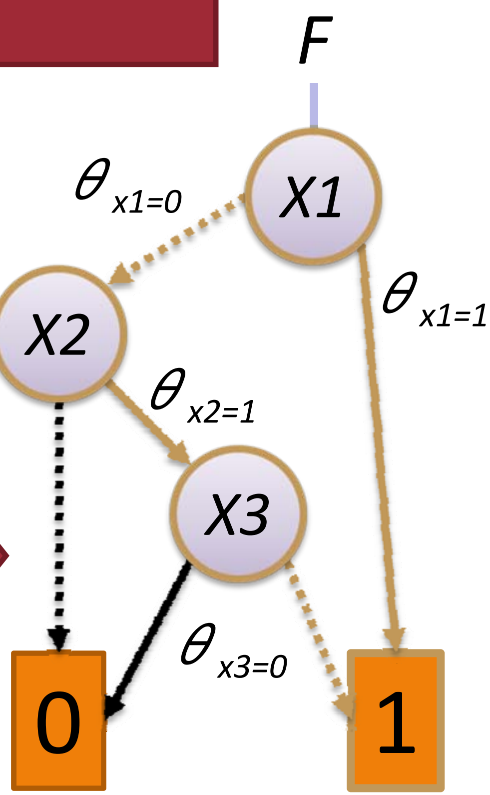
$P(F{=}f) = B[X1]$

$E[X1{=}1,F{=}f] = F_\theta[X1{=}1]\ \theta_{X1=1}\ B_\theta[1]$

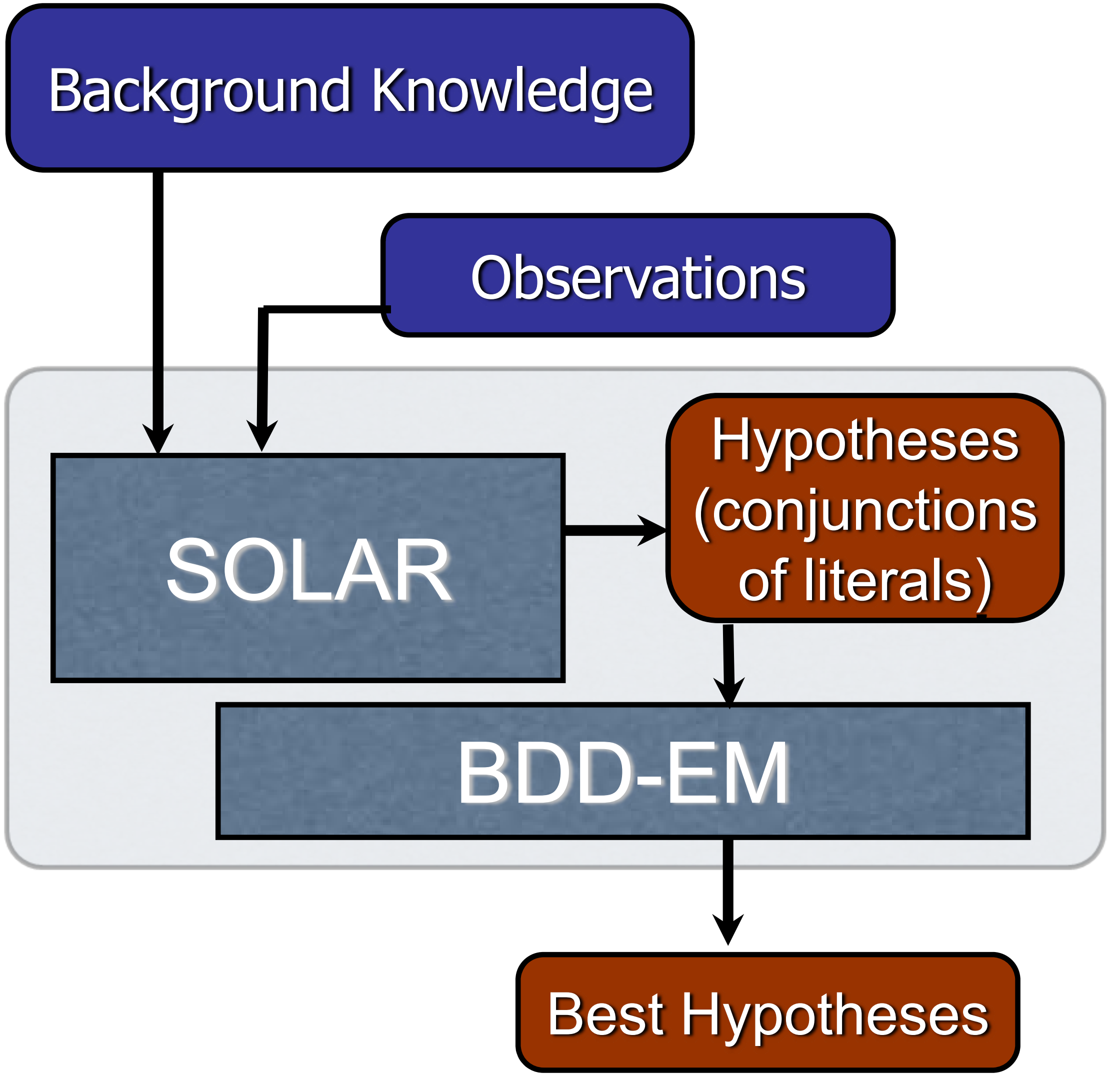
$E[X1{=}0,F{=}f] = F_\theta[X1{=}0]\ \theta_{X1=0}\ B_\theta[X2]$

M-step: Update probabilities

$$\theta_{X1=1} = \frac{E[X1{=}1,F{=}f]}{E[X1{=}1,F{=}f] + E[X1{=}0,F{=}f]}$$



The current abductive system



Hypothesis evaluation

- Given a numerous number of hypotheses,
- Which hypotheses are **most likely**?
- Statistical hypothesis selection
  - Probabilistic model specifying a distribution of hypotheses
  - Evaluation by the **BDD-EM** algorithm
  - Initial experiments for inhibitory effects using datasets for (Tamaddoni-Nezda *et al*., 2006)

Selecting the best explanations

- Many explanations: $H^{(1)}, H^{(2)}, \ldots, H^{(66)}$

$$B \wedge H^{(i)} \models O_1 \wedge O_2 \wedge \ldots$$

😐 All hypotheses logically explain the observations but we wish to choose the best one.

- We give probabilities to those atoms appearing in $H^{(i)}$ and B, and select $H^{(i)}$ that maximizes $P(B \wedge H^{(i)})$
- We learn probabilities (= parameters θ) from the disjunction of explanations as well as B, i.e.,

$$\theta^* = \mathrm{argmax}_\theta P((H^{(1)} \vee \ldots \vee H^{(k)}) \wedge B \mid \theta)$$

Proof-theoretic approximation

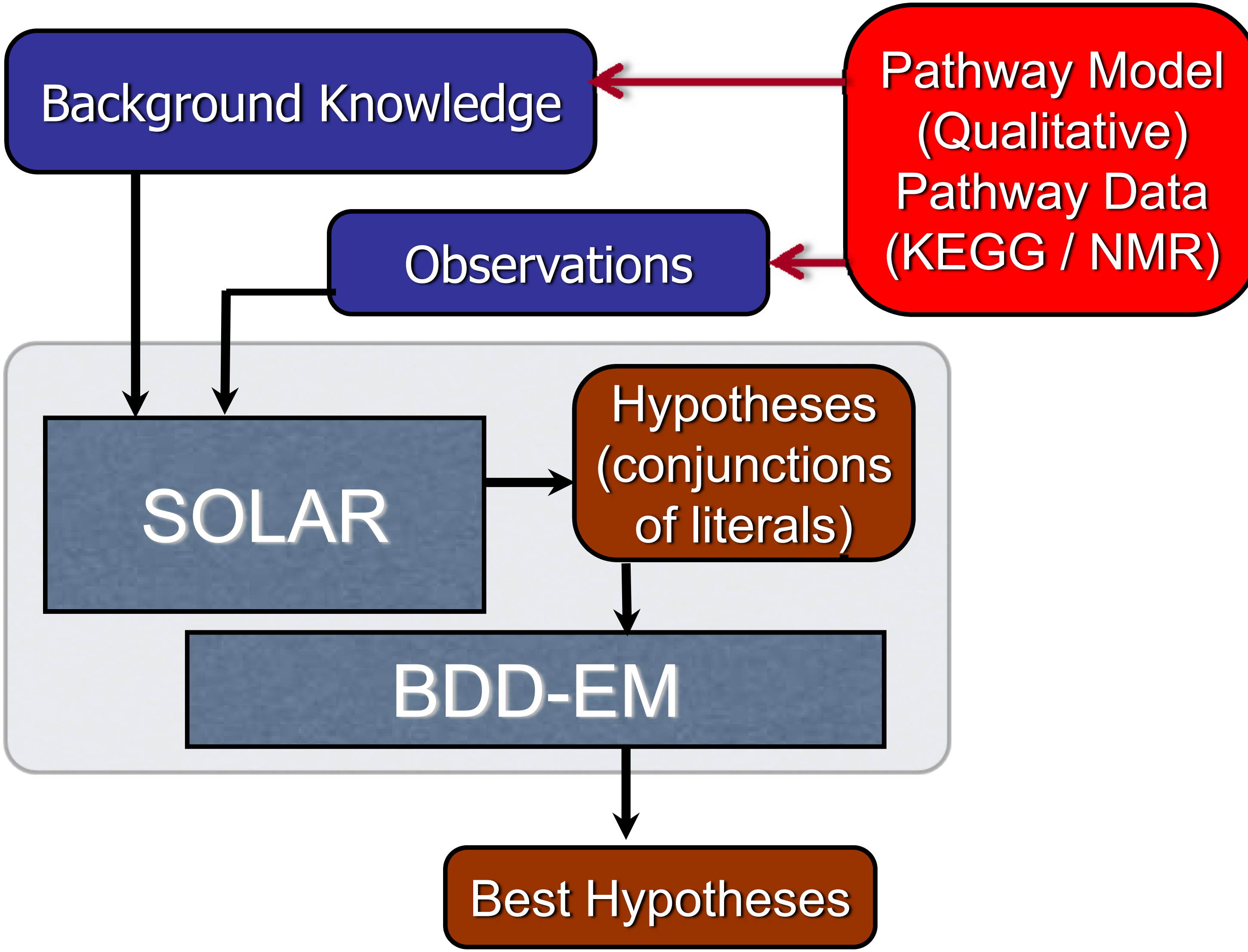
- Ground instances of *B* is infinite.
  - *O* : observations
  - $H^{(i)}$: an explanation abduced from *B* and *O*
  - $B^{(i)}$: subset of *B* relevant to $H^{(i)}$ and *O*, i.e., **proof**:

$$B^{(i)} \wedge H^{(i)} \vdash O$$

- We learn probabilities (= parameters θ) from the disjunction of explanations and their proofs, i.e.,

$$\theta^* = \mathrm{argmax}_\theta\, \boldsymbol{P}((H^{(1)} \vee \ldots \vee H^{(k)}) \wedge (B^{(1)} \wedge \ldots \wedge B^{(k)}) \mid \theta)$$

Experiments



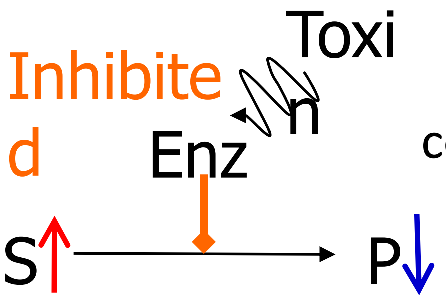
Prediction of inhibitory effects of a toxin
(Tamaddoni-Nezda *et al.*, *Machine Learning* 2006)

- Goal: Find inhibitions in a metabolic pathway
- Approach: Abduction (Inverse Entailment by SOLAR)

- Background Theory *B* :
  - Causal rules (4) and Integrity constraints (4)
  - Chemical reactions (76) in a metabolic network from KEGG
- Observations (Input) *E* :
  - Changes (up/down) of metabolites' concentrations (20*5=100)
- Hypothesis (Output) *H* :
  - A set (conjunction) of literals whose predicate is "inhibited"

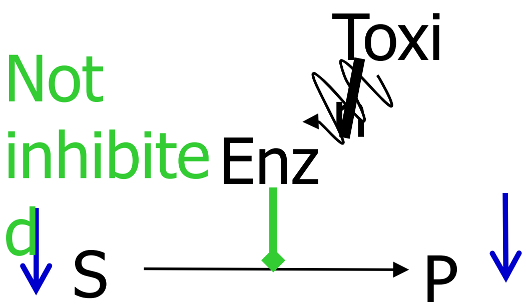
Logical modeling of Inhibition

(Tamaddoni-Nezda *et al.*, *Machine Learning* 2006)

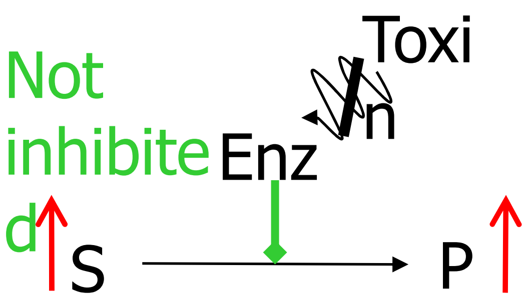


concentration(P, down) ← reaction(S, Enz, P), inhibited(Enz, S, P).

concentration(S, up) ← reaction(S, Enz, P), inhibited(Enz, S, P).



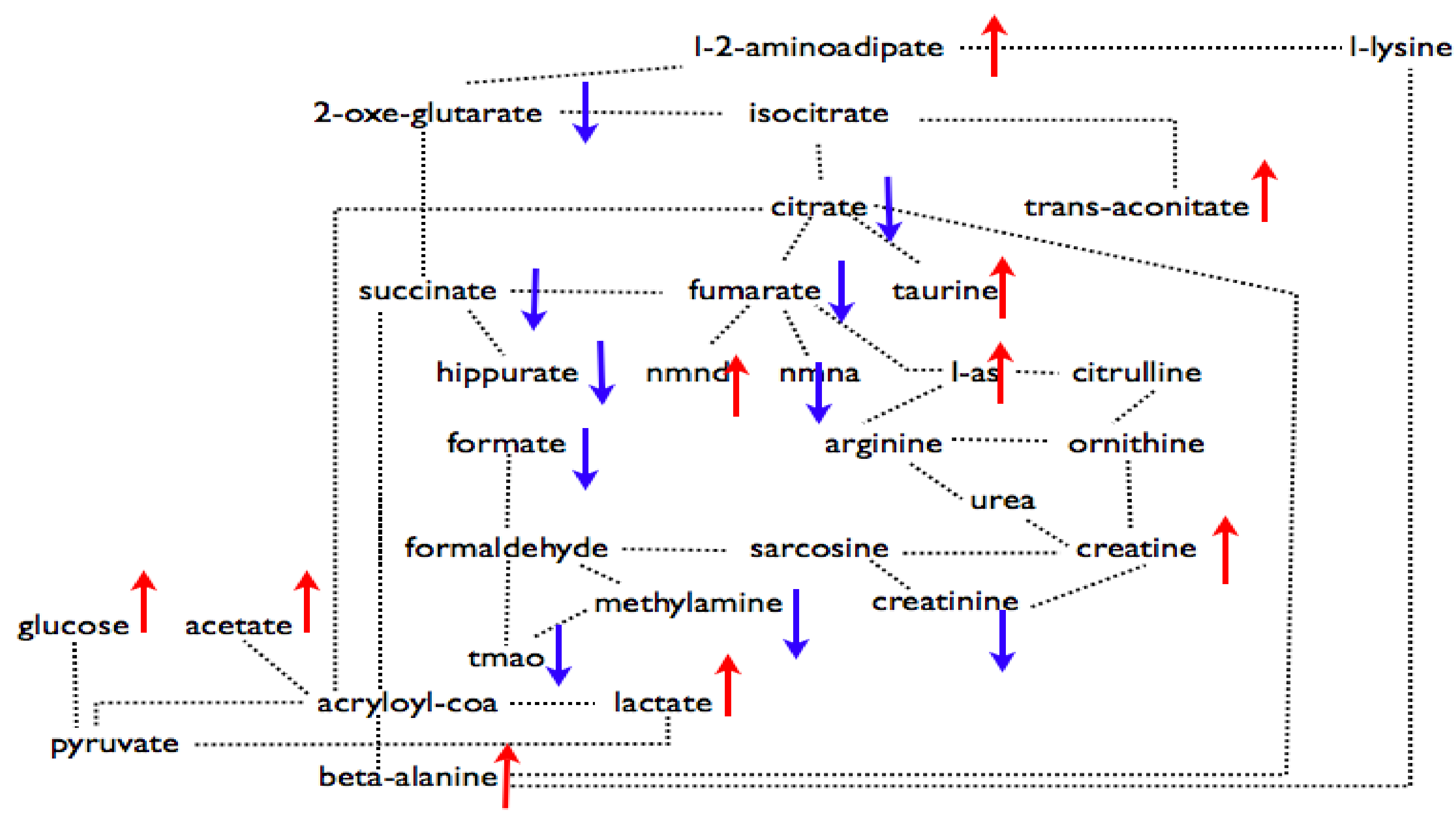
concentration(P, down) ← reaction(S, Enz, P),
¬inhibited(Enz, S, P), concentration(S, down).



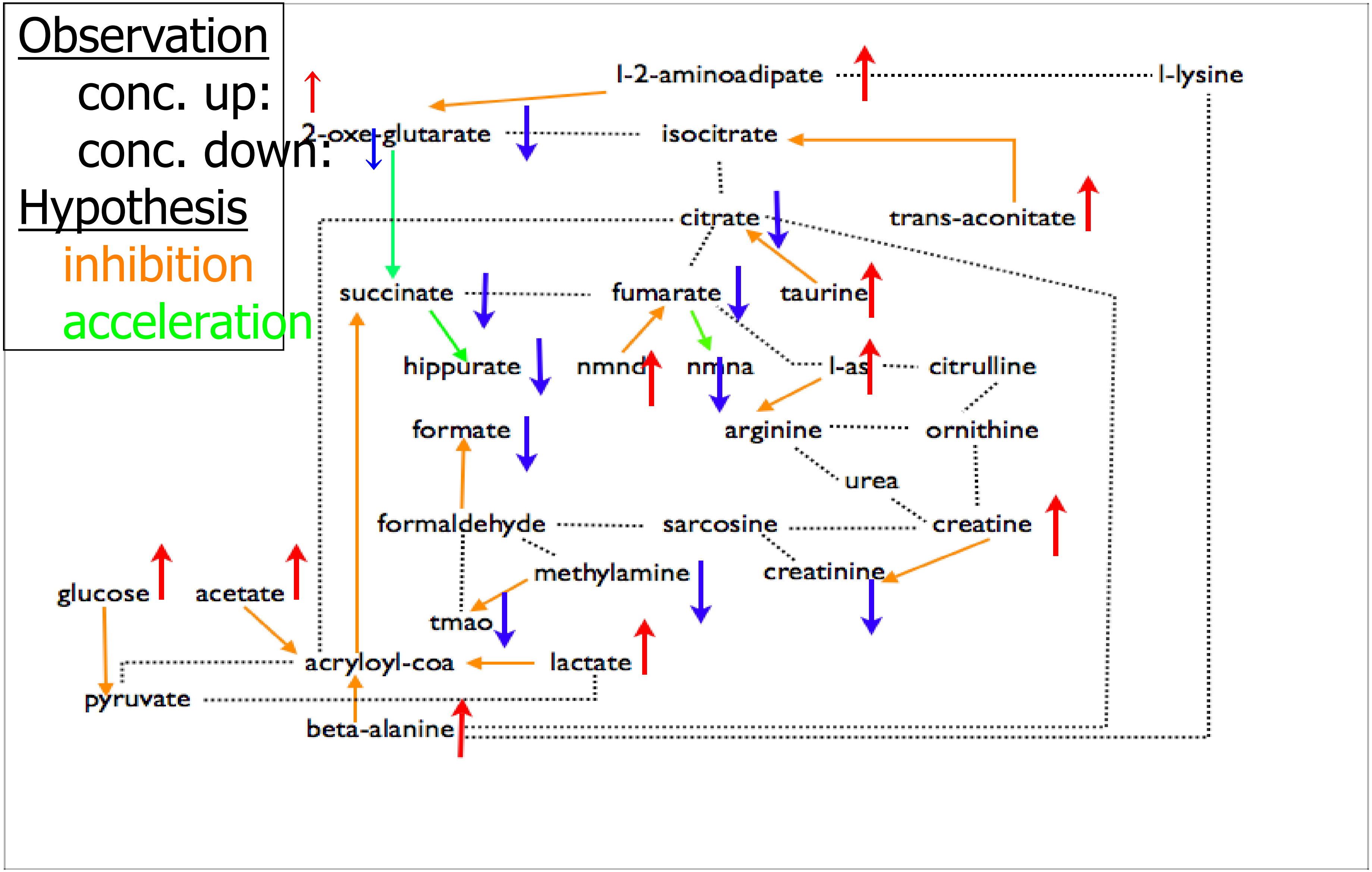
concentration(P, up) ← reaction(S, Enz, P),
¬inhibited(Enz, S, P), concentration(S, up).

Metabolic pathway representation

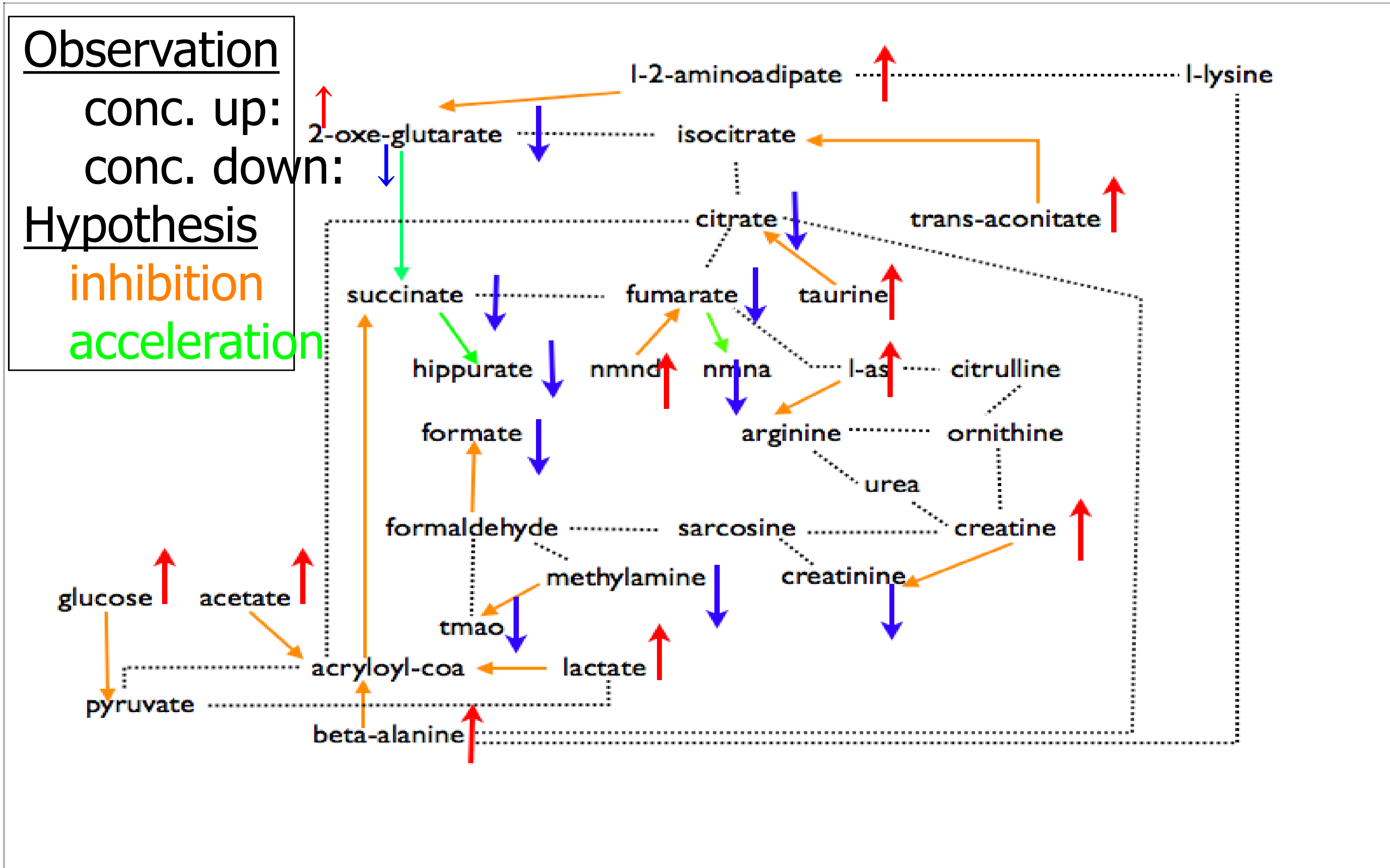
- Enzyme Reactions: 76
- Metabolites: 30
  - Extracellular: 20
  - Intracellular: 10



An output by SOLAR



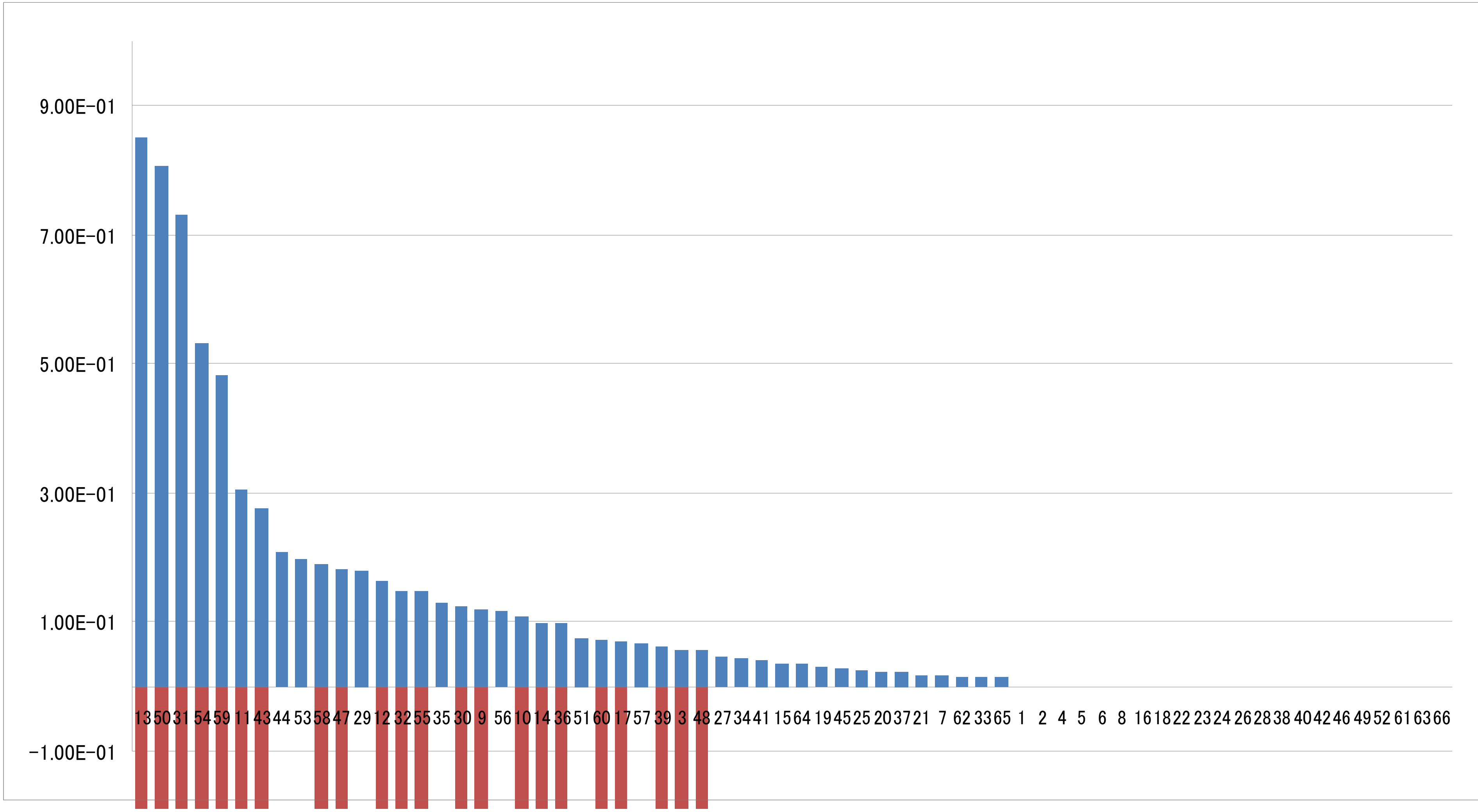
Another output by SOLAR



There are much more …

- SOLAR found 66 minimal explanations for 20 observations in Time = 8 hrs (and 5,145 minimal explanations in Time = 96 hrs).
- BDD-EM ranked all hypotheses according to their probabilities.
- The top 7 in Time = 8 hrs satisfy two desirable properties suggested by biologists.
- The worst 22 do not satisfy them.

Ranking of 66 hypotheses



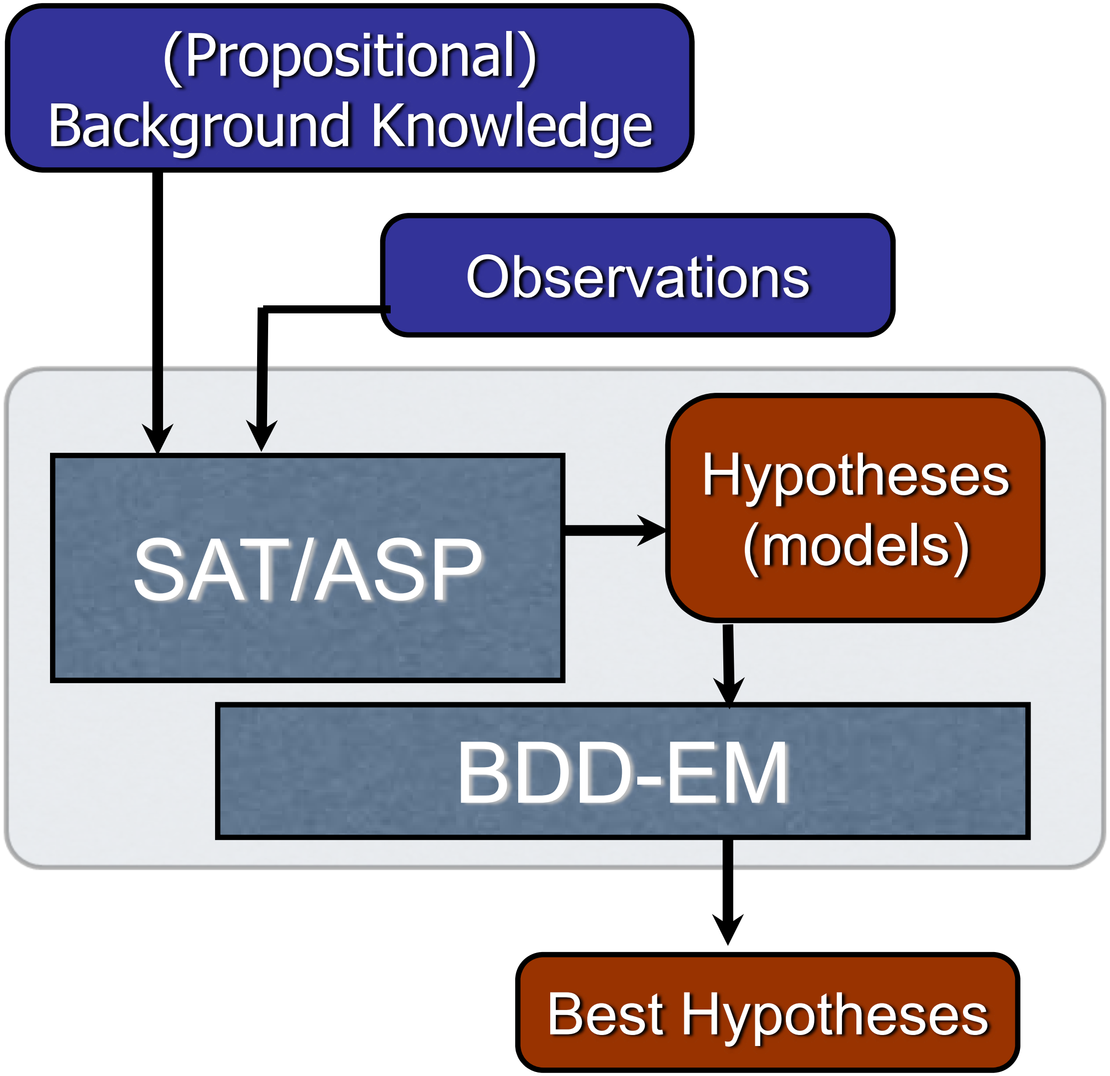
Statistics

	T=8	T=24	T=48	T=72	T=96
-df 5 -len 15	66	0	0	22	0
-df 5 -len 16	-	0	0	-	0
-df 5 -len 17	-	1638	3738	-	5145
SOLAR time	6m	2h34m	3h	5m	5h
SOLAR best	#13	#255	#1274	#10	#967
its prob	0.85	0.94	1.0-	1	1.0-
Progol best	#12	#14	#1043	#13	#865
its ranking	13	296	3713	20	862
its prob	0.16	0.003	0	0	0
ROBDD size	384	678	6226	335	2224
BDDEM time	5h49m	14h40m	134h7m	4h11m	50h48m

Related work

- PRISM (Sato & Kameya) – statistical abduction with EM
- ProbLog (De Raedt *et al*., 2007) – prob comp on BDD
- (Simon & del Val, 2001) – consequence finding on ZBDDs
- (Hsu *et al*., 2007) – EM for finding a solution in CSP
- Abduction in Systems Biology: (Zupan *et al*., 2003), (King *et al*., 2004), (Tran *et al*., 2005) – incomplete hypothesis finding, no statistical evaluation of hypotheses

Yet another system

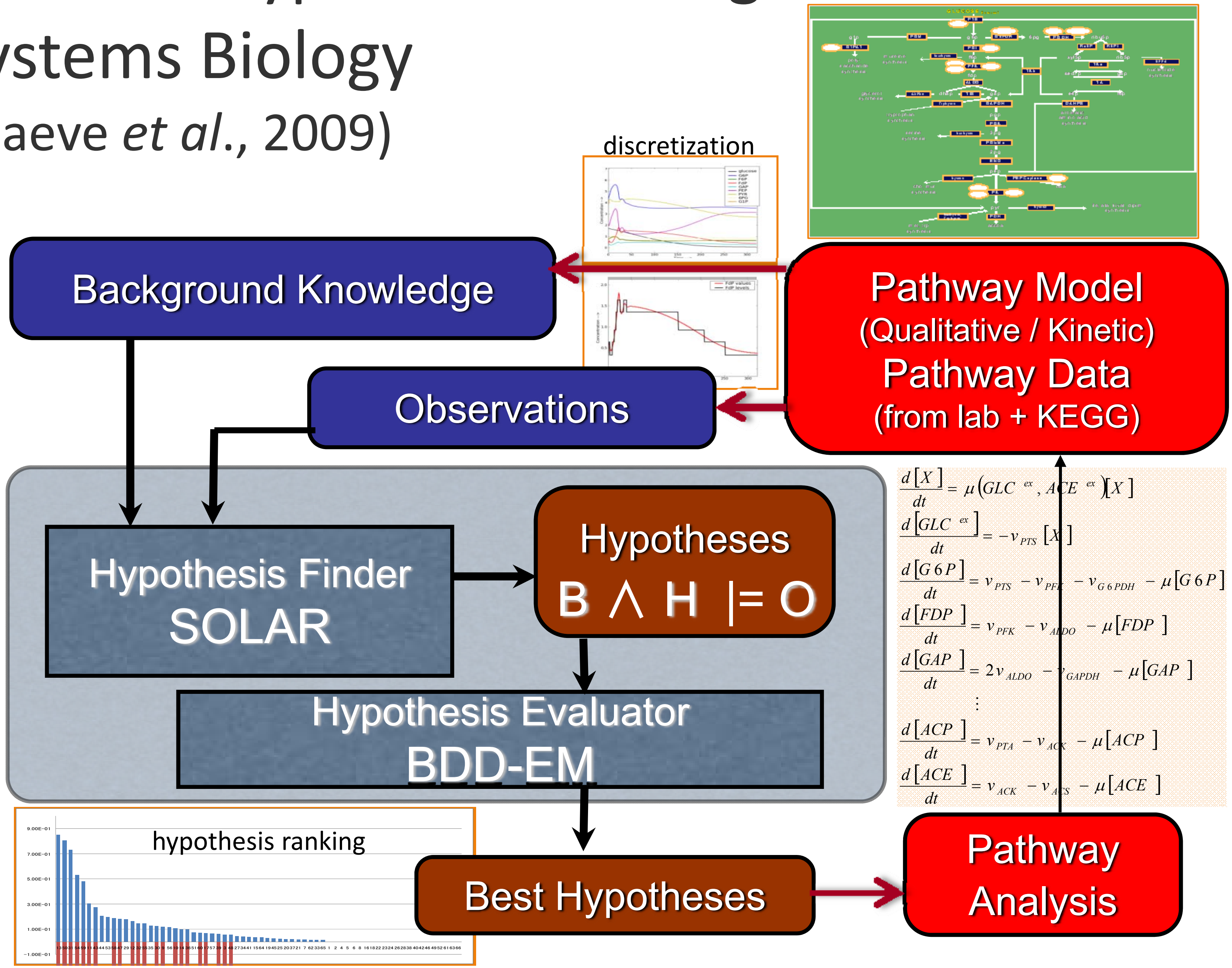


Conclusion

- A novel abductive architecture with
  - complete hypothesis generation (SOLAR)
  - statistical hypothesis evaluation (BDD-EM)
- Allows full clausal theories for background knowledge
  - cyclic dependencies
  - disjunctions
- An alternative way to select best hypotheses: BDD-EM with conditional distribution (Sato *et al*., *ILP* 2009)
- Application to hypothesis finding in Systems Biology

Automated hypothesis-finding in Systems Biology

(Synnaeve *et al.*, 2009)



Discovering Rules by Meta-level Abduction on SOLAR

Katsumi Inoue
National Institute of Informatics
Hidetomo Nabeshima
University of Yamanashi

Koichi Furukawa
Ikuo Kobayashi
Keio University

adapted from presentation at ILP'09

In this talk

- We propose a method to **abduce** rules, which enables us to infer hypotheses
  - representing *multiple* missing causal relations,
  - accounting for *multiple* observations simultaneously,
  - containing **new predicates**.
- The method provides a new way of induction based on **full-clausal** abduction.
- Combination of rule abduction and fact abduction is possible by way of **conditional query answering**.
- A motivating example is taken from **cognitive modeling**, but the method can be applied to **scientific discovery** from **network data**, e.g., biochemical pathways.

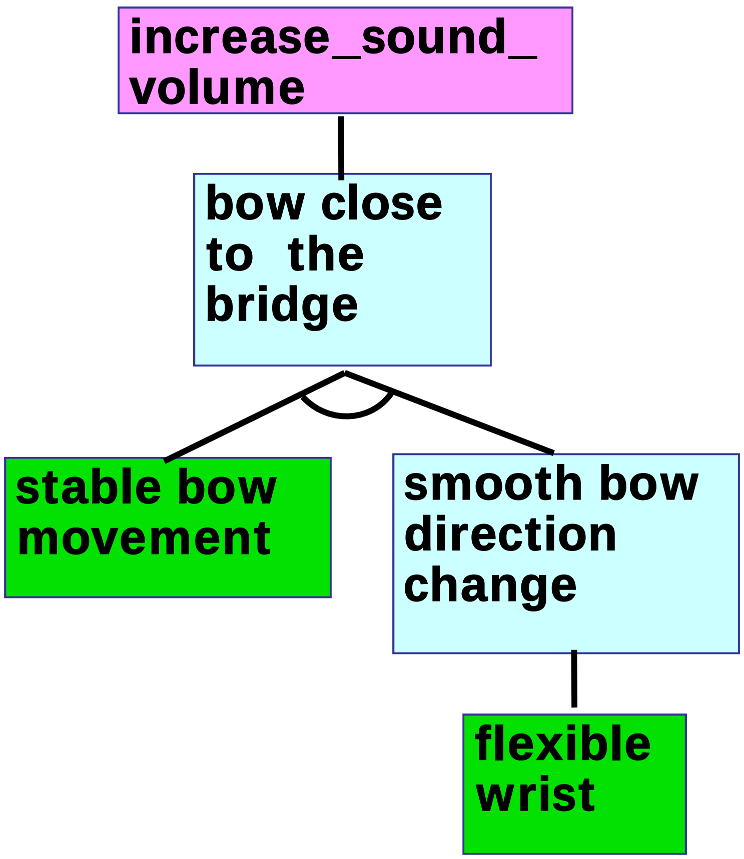
Motivation

- Prof. F experienced sudden **skill improvement** of cello playing after his final lecture concert.
- It was brought by simply keeping his right arm shut, that is, to keep his elbow close to the body side.
- This devise has increased the sound volume. Moreover, it keeps the bowing stable and maximum bow usage.
- But, *any finding cannot be applied unless it is explained*.
- *Reproduction* of good skill also requires **explanation**, which makes the skill *tolerant to situation changes*.
- The process of explanation will further lead to another important finding. This is the same as **scientific discovery**.
- Prof. F calls this "***knack discovery***", and tried to formulate it.

Explaining skill improvement (1)

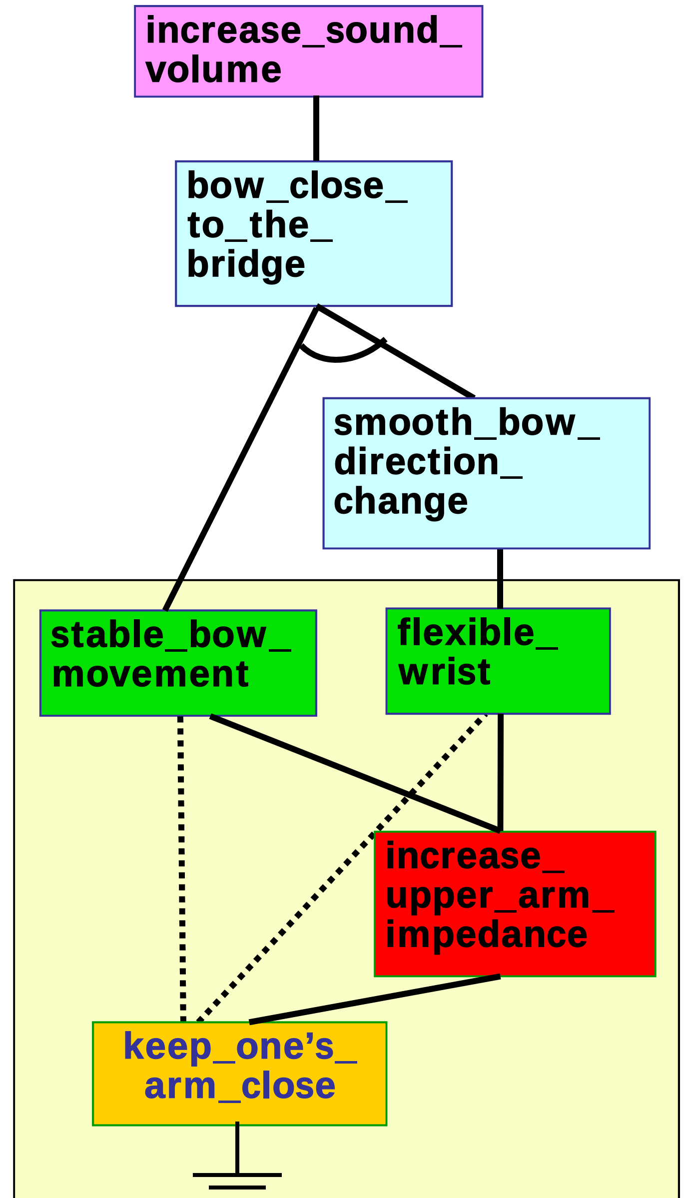
- **F's skill improvement** was brought by *keeping his arm close* to the body side, which resulted in *increase of the sound volume*.
- **Background knowledge**:

1. To increase_sound_volume, we need bow_close_to_the_bridge.
2. To keep bow vibration, we need (1) stable_bow_movement, and (2) smooth_bow_direction_change, which needs flexible_wrist.



Explaining skill improvement (2)

- The ***goal*** is increase_sound_volume.
- This goal has been empirically achieved by the ***stimulus*** keep_one's_arm_close.
- With the background knowledge, keep_one's_arm_close should cause two states: (1) stable_bow_movement and (2) flexible_wrist.
- However, these relations do not directly hold. Instead, introduction of the ***hidden attention***: increase_upper_arm_impedence can fill the gap of inference.



Logical Representation

increase_sound_volume ← bow_close_to_bridge.

**bow_close_to _bridge ← stable_bow_movement ∧
smooth_bow_direction_change.**

smooth_bow_direction_change ←flexible_wrist.

**stable_bow_movement ←
increase_upper_arm_impedance.**

flexible_wrist ← increase_upper_arm_impedance.

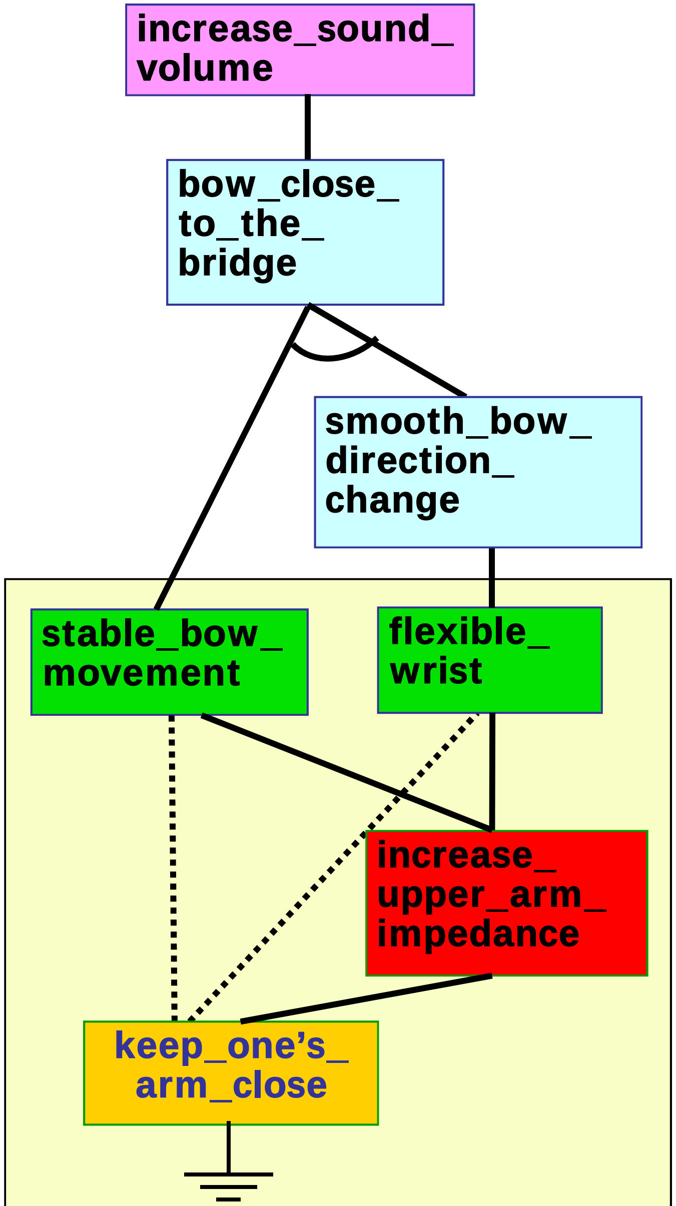
increase_upper_arm_impedance ← keep_arm_close.

Filling the gap of proofs

- In this program, the goal:

  ?- increase_sound_volume.

  will give the proof in the right.
- The part is augmented by introducing the hidden attention .
- The gap filling is the task of **abduction**.



Causality

- To explain empirical rules, we need **causal chains**.
- Causality can be represented in first-order predicate logic.
- Two predicates:

1. **connected(X,Y)**: event X is *directly caused by* event Y.
2. **caused(X,Y)**: there is a *causal chain* from event Y to event X.

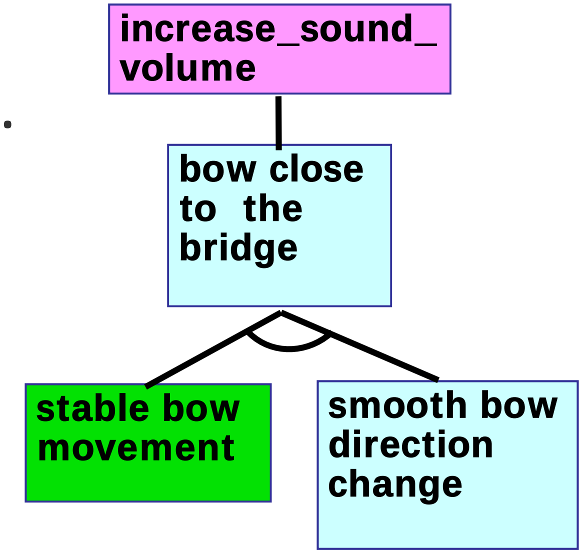
$$\text{caused}(X,Y) \leftarrow \text{connected}(X,Y).$$
$$\text{caused}(X,Y) \leftarrow \text{connected}(X,Z) \wedge \text{caused}(Z,Y).$$

Object and meta level representation

- **Object domain (object level)**

increase_sound_volume ← bow_close_to_bridge.

bow_close_to _bridge ←
stable_bow_movement ∧
smooth_bow_direction_change.

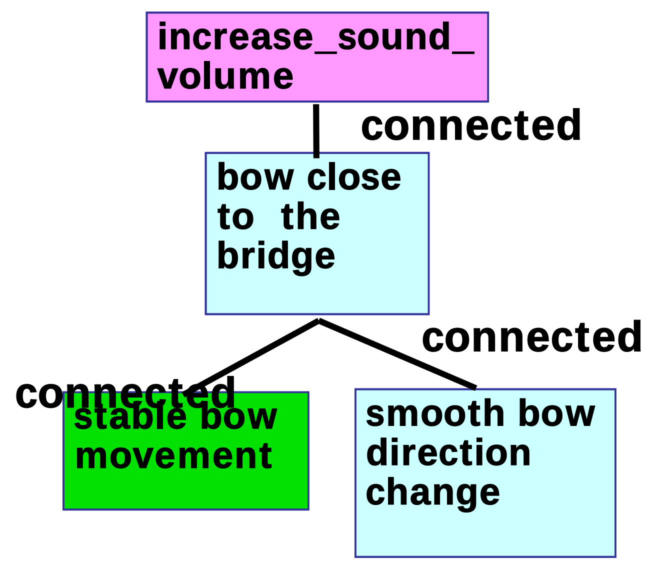


- **Causal relations (meta level)**

connected(increase_sound_volume,
bow_close_to_bridge).

connected(bow_close_to_bridge,
stable_bow_movement).

connected(bow_close_to_bridge,
smooth_bow_direction_change).



- **Each *literal* in the object level is represented as a *term* in the meta level.**
- No "AND" connective here...

Abductive Reasoning

- Abduction augments sufficient conditions missing in the premises (or background knowledge) to enable a derivation of the observation.
- This fills the gap in a proof of the observation from the premises.
- Inferred conditions are called **hypotheses** or **explanations**.

Problem setting

1. **Rule abduction**:

To fill the gap between

keep_one's_arm_close and

increase_sound_volume,

we augment the rule:

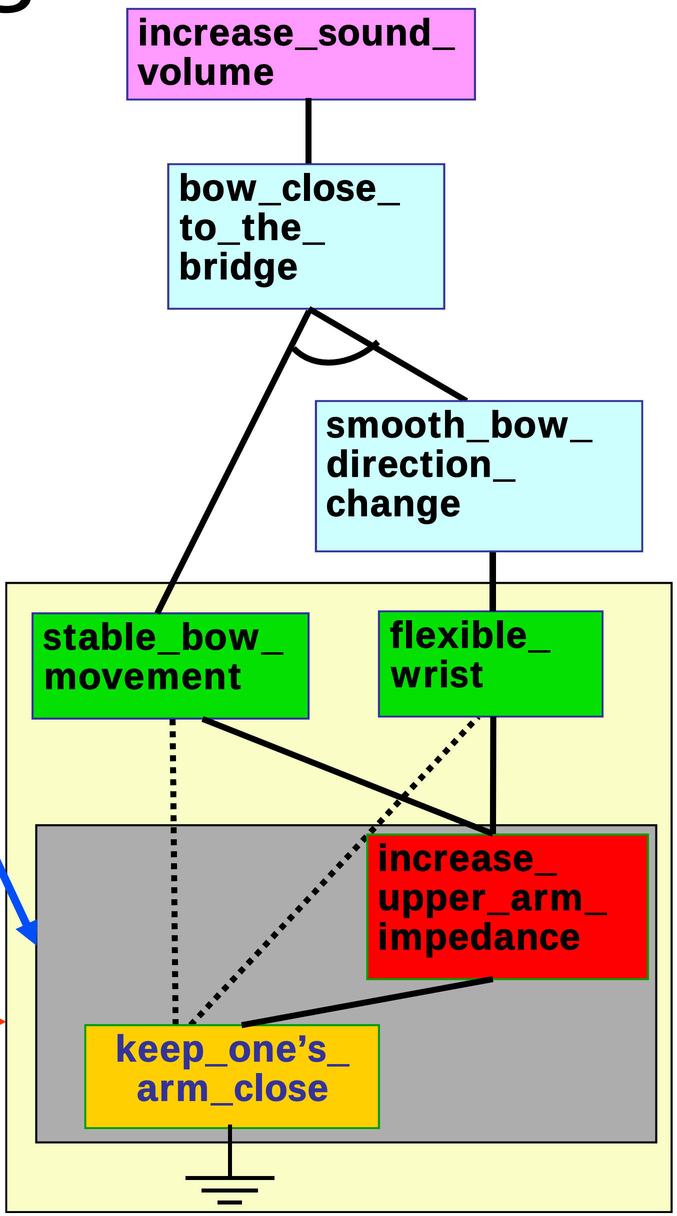
inrease_upper_arm_impedence ← keep_one's_arm_close.

2. **Predicate invention**:

The hidden attention

increase_upper_arm_impedence

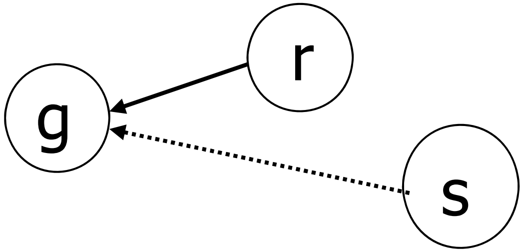
must be found. .



Formalizing rule abduction

- *g*: a **goal,** *s*: an **input,** *r*: a (hidden) node

B: *connected(g, r).*
$\leftarrow$ *connected(g, s).*



That is, *g* is directly caused by *r*, but *g* is ***not*** directly caused by *s*.

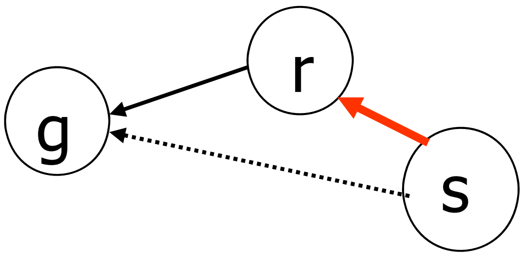
- *g* is not directly caused by *s*, but we know that there is a causal chain to *g* from *s*. This is given by an observation:

G: *caused(g, s).*

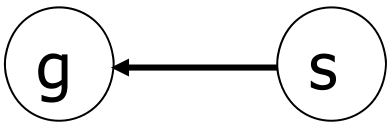
- SOLAR computes a hypothesis

H: *connected(r, s),*

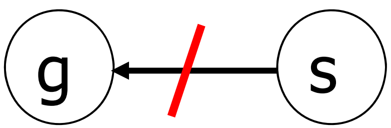
given the abducibles {*connected(_,_)*}.



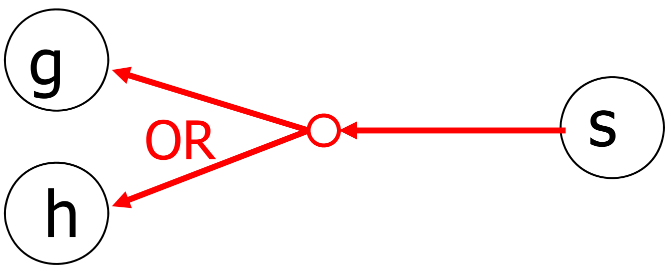
Representing logical connectives



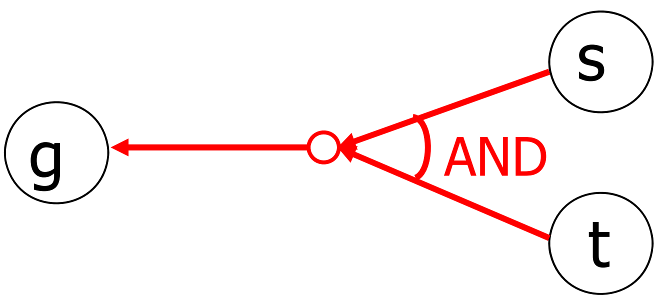
connected(g, s).



$\neg$*connected(g, s).*



connected(g, s) $\vee$ *connected(h, s)*



connected(g, s) $\vee$ *connected(g, t)*

Rule abduction example

B: connected(inc_sound, bow_close_to_the_bridge).

connected(bow_close_to_the_bridge, stable_bow_movement) $\vee$
connected(bow_close_to_the_bridge, smooth_bow_direction_change).

connected(smooth_bow_direction_change, flexible_wrist).

connected(stable_bow_movement, increase_upper_arm_impedance).

connected(flexible_wrist, increase_upper_arm_impedance).

$\leftarrow$ connected(inc_sound, keep_arm_close).

G: caused(inc_sound, keep_arm_close).

abducible: connected(_,_).

H: connected(increase_upper_arm_impedance, keep_arm_close).

Obtained hypothesis

H:

connected(increase_upper_arm_impedance,

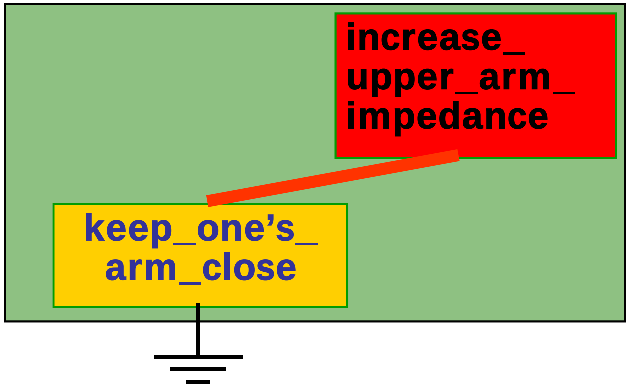
keep_arm_close).

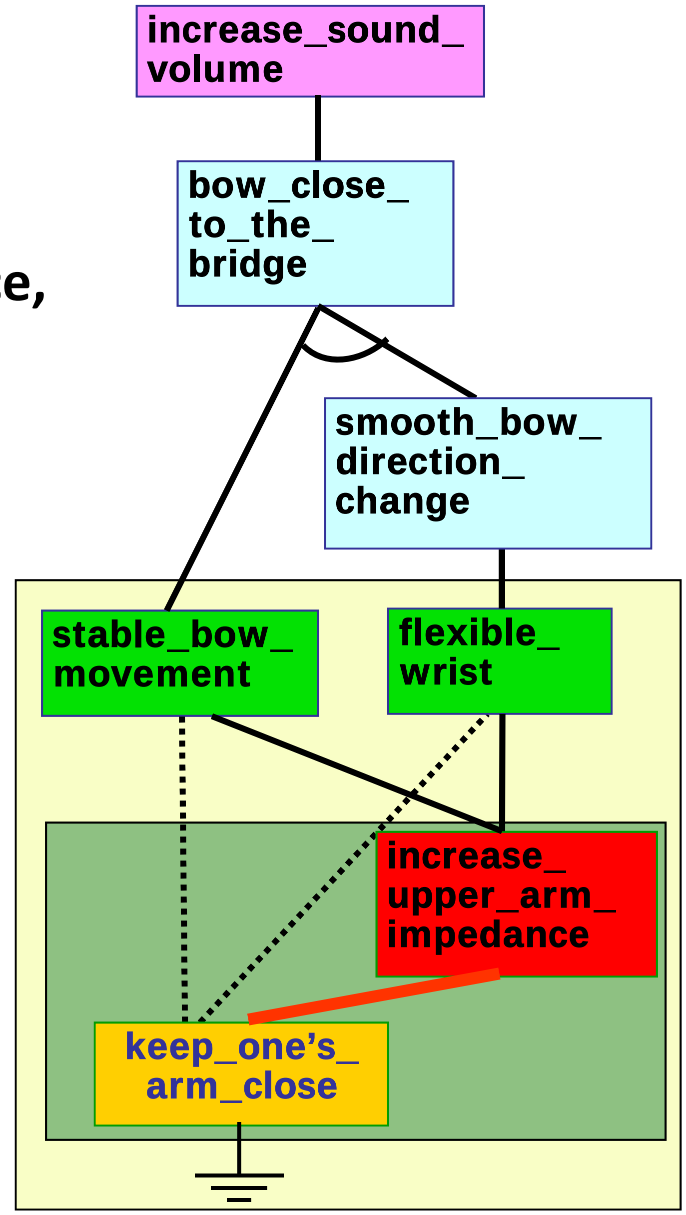
- In the object level, this means:

increase_upper_arm_impedance

← keep_arm_close.

- The rule:





Predicate invention

- Predicate invention consists of the 2 steps:

1. Fill the gap in a proof of a causal chain by introducing a new node ➔ abduction by SOLAR producing **existentially quantified hypotheses**
2. Give the meaning of the introduced node ➔ identification of the new predicate

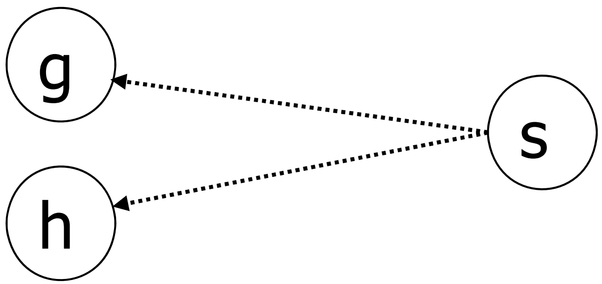
Formalizing node introduction

- *g*: a goal, *s*: an input, *r*: a (hidden) node

B: ← *connected(g, s).*
← *connected(h, s).*

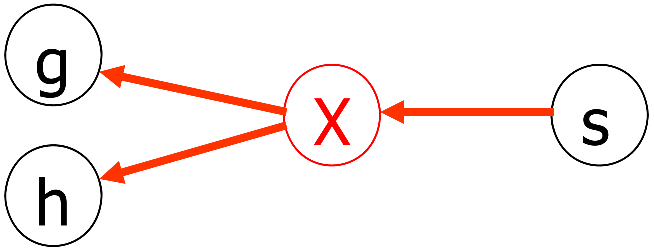
That is, there are no direct causal relation from s to g and from h to s, but there are causal chains as the observations:

G: *caused(g, s).*
caused(h, s).



- Given the abducibles {*connected(_,_)*}, SOLAR generates a hypothesis **H:**

$\exists X.$ (*connected(g, X)* ∧ *connected(h, X)* ∧ *connected(X, s)*).

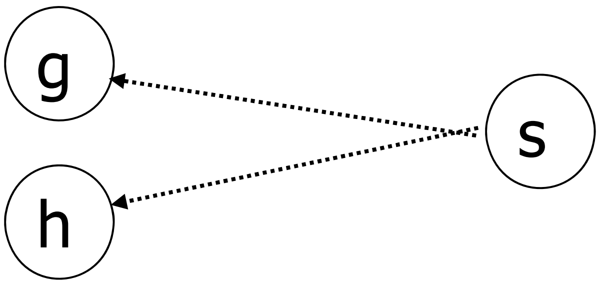


- Variable *X* represents a newly introduced node.

Representing different structures

B: ← *connected(g, s).*
← *connected(h, s).*

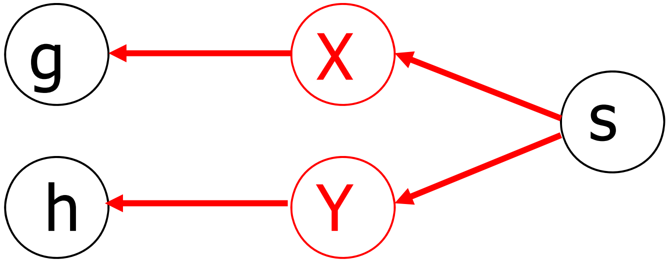
G: *caused(g, s).*
caused(h, s).



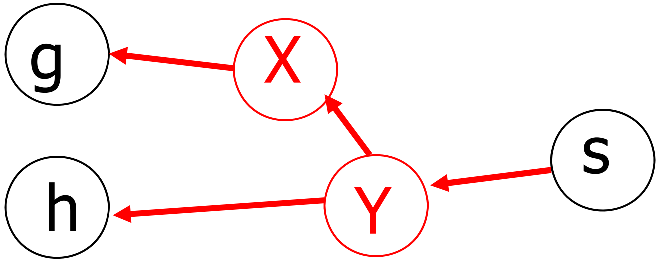
Abducibles: {*connected(_,_)*}.

H with 2 intermediate nodes:

$\exists X \exists Y.$ (*connected(g, X)* ∧ *connected(h, Y)*
∧ *connected(X, s)* ∧ *connected(Y, s)*).



$\exists X \exists Y.$ (*connected(g, X)* ∧ *connected(h, Y)*
∧ *connected(X, Y)* ∧ *connected(Y, s)*).



Correctness of meta-level abduction

Lemma:

Let *λ(B) be the theory obtained by replacing every connected(g, s) appearing in B with the formula (g ← s).*

If B $\models$ *caused(g, s) then λ(B)* $\models$ *(g ← s).*

Theorem: *Suppose the observation caused(g, s). If H is an abductive explanation of caused(g, s) with respect to B and* Γ_M *= {connected(_,_)}, then λ(H) is a hypothesis s.t.*

- *λ(B) ∪ λ(H)* $\models$ *(g ← s), and*
- *λ(B) ∪ λ(H) is consistent.*

Correctness of meta-level abduction

Theorem:

Suppose the background knowledge K in the object level, and let C(K) be the meta-theory representing the causal graph associated with K, and define that

$$\tau(K) = C(K) \cup \{\ caused(X,Y) \leftarrow connected(X,Y).$$
$$caused(X,Y) \leftarrow connected(X,Z) \land caused(Z,Y).\}.$$

If g is reached from s in the causal graph of K by augmenting a set E of direct causal relations, then C(E) is an abductive explanation of caused(g, s) with respect to $\tau(K)$ and Γ_M.

Application to knack discovery

B: connected(inc_sound, bow_close_to_the_bridge).

connected(bow_close_to_the_bridge, stable_bow_movement) $\vee$
connected(bow_close_to_the_bridge, smooth_bow_direction_change).

connected(smooth_bow_direction_change, flexible_wrist).

$\leftarrow$ connected(inc_sound, keep_arm_close).

$\leftarrow$ connected(stable_bow_movement, keep_arm_close).

$\leftarrow$ connected(smooth_bow_direction_change, keep_arm_close).

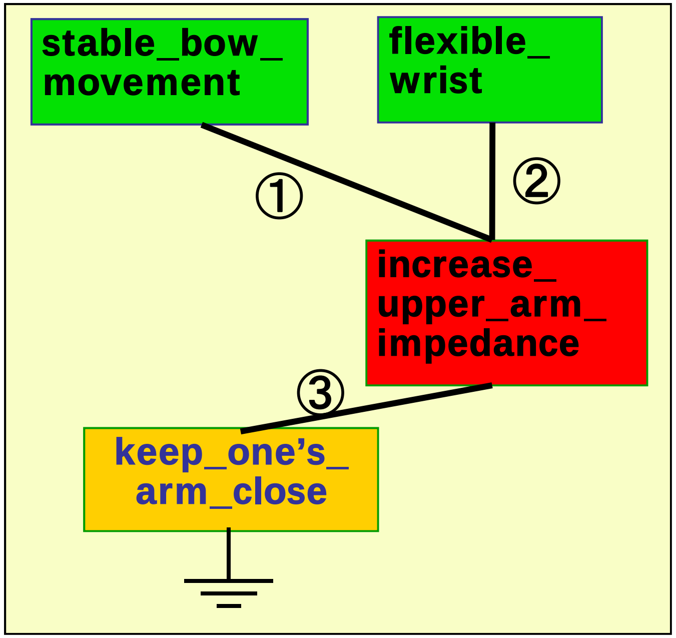
G: caused(inc_sound, keep_arm_close).

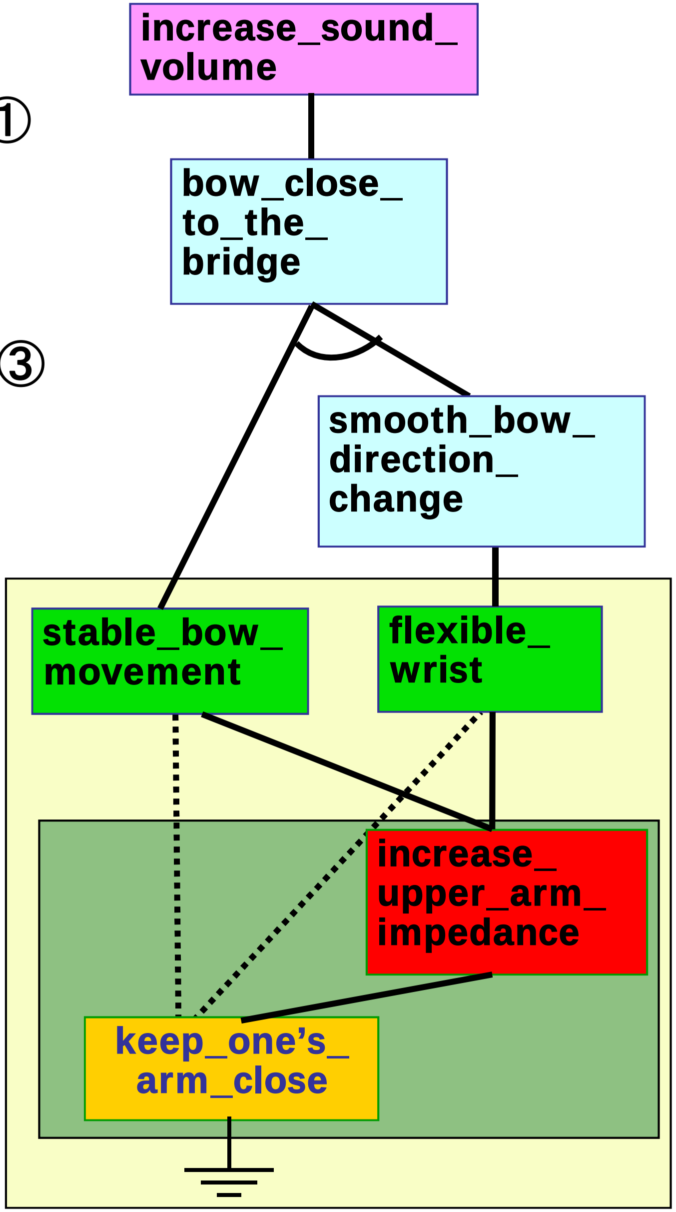
- SOLAR generates 52 hypotheses when the maximum search depth is 15 and the maximum length of produced clauses is 5. One of them is:

  ∃X. (connected(stable_bow_movement, X)
  ∧ connected(flexible_wrist, X)
  ∧ connected(X, keep_arm_close)).

The obtained hypothesis

H: ∃X. (connected(stable_bow_movement, X) ①
∧ connected(flexible_wrist, X) ②
∧ connected(X, keep_arm_close)). ③





Identifying new nodes

- An obtained new node is meaningful if the direct causal relations between this node and other neighbors are conceivable.
- A new node corresponds to a predicate in the object level.
- This new predicate may be unknown.
- Identification of a new predicate is nothing but predicate discovery.

Identifying new predicate

H: $\exists$ X. (connected(stable_bow_movement, X)
$\wedge$ connected(flexible_wrist, X)
$\wedge$ connected(X, keep_arm_close)).

- What does this X mean?
- *Anatomical clues* are: armrest, wrist, forearm, elbow, upper arm, brachial muscles (biceps, triceps) as parts of human body.
- *Conditions associated with body parts* are: positions and postures of parts, activity of muscles, velocity and acceleration of movement.
- Select a candidate from these, then substitute X with it, and check its validity.
- The role of *flash of inspiration*

Abducing facts

New axioms:

caused(X, X) ← abd(X). *% for abducibles*

caused(X, Y) ← connected(X, Y).

caused(X, Y) ← connected(X, Z) ∧ caused(Z, Y).

The top clause:

← caused(g, X) ∧ abd(X).

<u>*Note*</u>*: abd* plays the role of an *answer predicate.*

An integrity constraint that *p* and *q cannot hold simultaneously:*

← caused(p, X) ∧ caused(q, Y) ∧ abd(X) ∧ abd(Y).

Abducing facts and rules

- Abducing facts is nothing but **answer extraction**.
- Abducing facts and rules is then **conditional query answering**.

Correspondence between object-level inference and meta-level consequence finding

object-level inference	top clause in SOLAR	production field
proving rules	$\neg caused(g, s)$	none
abducing facts	$\neg caused(g,X) \lor \neg abd(X)$	$\neg abd(f1), \ldots, \neg abd(fn)$
predicting facts	$\neg caused(X, s) \lor ans(X)$	$ans(\_)$
predicting rules	none	$caused(\_,\_)$
abducing rules	$\neg caused(g, s)$	$\neg connected(\_,\_)$
abducing rules and facts	$\neg caused(g,X) \lor \neg abd(X)$	$\neg connected(\_,\_)$ $\neg abd(f1), \ldots, \neg abd(fn)$
predicting conditional facts	$\neg caused(X, s) \lor ans(X)$	$\neg connected(\_,\_), ans(\_)$
predicting conditional rules	none	$\neg connected(\_,\_),$ $caused(\_,\_)$

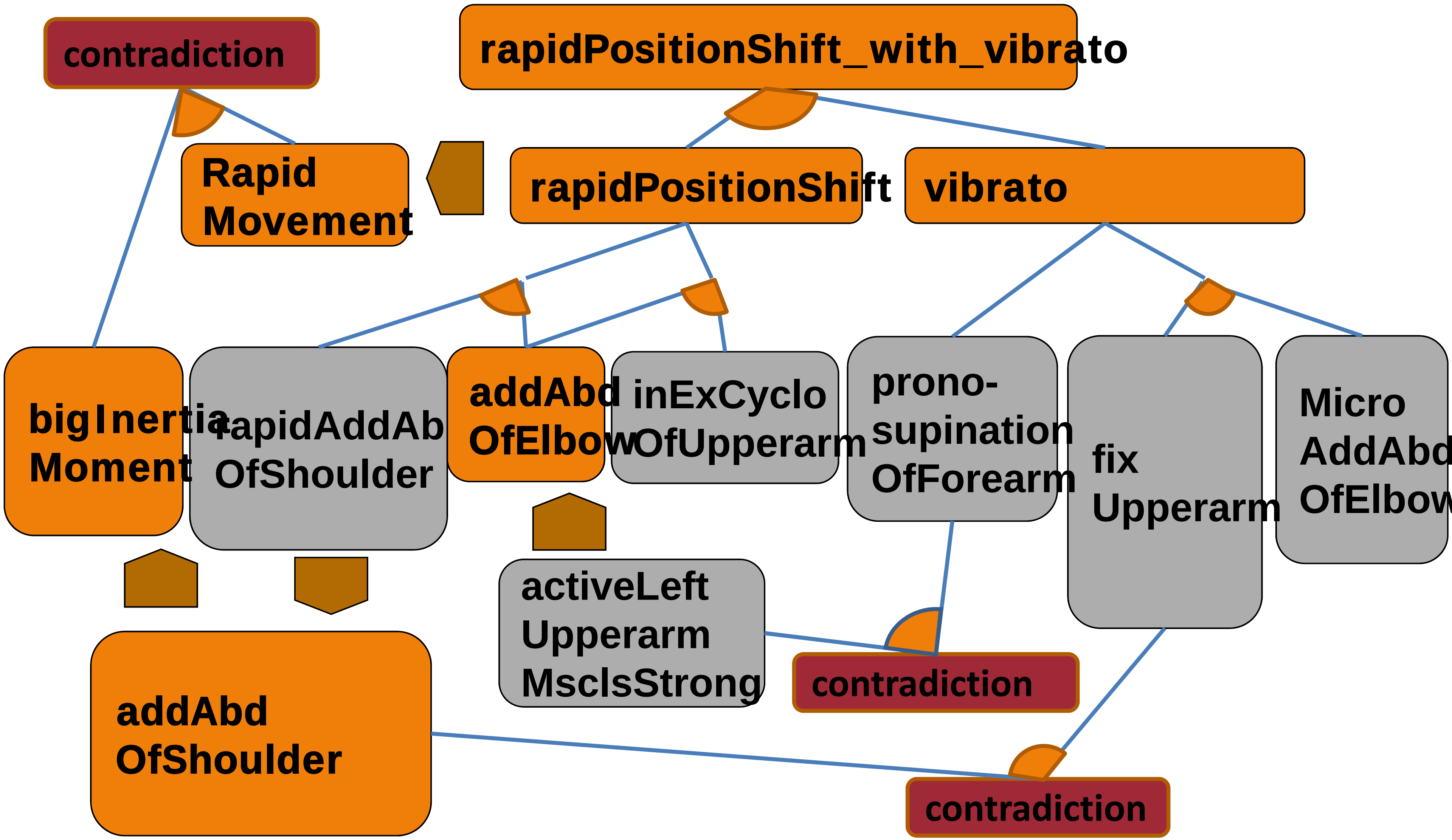
Physical skill discovery by abduction
(Furukawa & Kobayashi, 2008)

- Goals: Position shift with continuous vibrato
- Motion Integrity Constraints :
  1. "Fixing the upper arm" and "Adduction/abduction of the upper arm" contradict each other.
  2. "Rapid movement" and "Big moment of inertia" contradict each other.
- abducibles:

  rapid_add_abd_of_Shoulder,
  active_upper_arm_mscls_strong,

  in_exCyclo_of_upper_arm, fix_upper_arm,
  pronosupination_of_Forearm, micro_add_abd_of_Elbow

Scheme of Example Program



Object-level representation

% Causal rules
rapidPositionShift_with_vibrato ← rapidPositionShift ∧ vibrato.
rapidPositionShift ← rapidAddAbdOfarm ∧ flexExtOfElbow.
rapidPositionShift ← inExCycloOfUpperarm ∧ pronosupinationOfForearm.
vibrato ← fixUpperarm ∧ microFlexExtOfElbow.
vibrato ← pronosupinationOfForearm.
flexExtOfElbow ← activeLeftUpperarmMsclsStrong.

% Integrity constraints
← pronosupinationOfForearm ∧ activeLeftUpperarmMsclsStrong.

% Abducible predicates
abducible(rapidAddAbdOfarm). abducible(inExCycloOfUpperarm).
abducible(microFlexExtOfElbow). abducible(pronosupinationOfForearm).
abducible(fixUpperarm). abducible(activeLeftUpperarmMsclsStrong).

Meta-level representation

% Causal graph theory
connected(rapidPositionShift_with_vibrato, rapidPositionShift)
 ∨ connected(rapidPositionShift_with_vibrato, vibrato)
connected(rapidPositionShift, rapidAddAbdOfarm)
 ∨ connected(rapidPositionShift, flexExtOfElbow).
connected(rapidPositionShift, inExCycloOfUpperarm)
 ∨ connected(rapidPositionShift, pronosupinationOfForearm).
connected(vibrato, fixUpperarm) ∨ connected(vibrato, microFlexExtOfElbow).
connected(vibrato, pronosupinationOfForearm).
connected(flexExtOfElbow, activeLeftUpperarmMsclsStrong).

% Integrity constraints
← caused(pronosupinationOfForearm, X) ∧ abd(X)
 ∧ caused(activeLeftUpperarmMsclsStrong, Y) ∧ abd(Y).

% Top clause: C1
← caused(rapidPositionShift_with_vibrato, X) ∧ abd(X).

Production field

$\mathcal{P}1 = < -\mathcal{A}bd,\ |C| \leq 5 >$

$-\mathcal{A}bd$:

¬ abd(rapidAddAbdOfarm). ¬ abd(inExCycloOfUpperarm).
¬ abducible(microFlexExtOfElbow). ¬ abd(pronosupinationOfForearm).
¬ abd(fixUpperarm). ¬ abd(activeLeftUpperarmMsclsStrong).

The unique new characteristic clause:

¬ abd(inExCycloOfUpperarm) ∨ ¬ abd(pronosupinationOfForearm).

Meta-level representation (modified)

% Causal graph theory
connected(rapidPositionShift_with_vibrato, rapidPositionShift)
  ⋁ connected(rapidPositionShift_with_vibrato, vibrato)
connected(rapidPositionShift, rapidAddAbdOfarm)
  ⋁ connected(rapidPositionShift, flexExtOfElbow).
connected(rapidPositionShift, inExCycloOfUpperarm)
  ⋁ connected(rapidPositionShift, pronosupinationOfForearm).
connected(vibrato, fixUpperarm) ⋁ connected(vibrato, microFlexExtOfElbow).
%connected(vibrato, pronosupinationOfForearm).
connected(flexExtOfElbow, activeLeftUpperarmMsclsStrong).

% Integrity constraints
← caused(pronosupinationOfForearm, X) ⋀ abd(X)
  ⋀ caused(activeLeftUpperarmMsclsStrong, Y) ⋀ abd(Y).

% Top clause: C1
← caused(rapidPositionShift_with_vibrato, X) ⋀ abd(X).

Production field

$\mathcal{P}2$ = < -$\mathcal{Abd}$ ∪ {¬connected(_,_)}, |C| ≤ 5 and |C∩{¬connected(_,_)}| ≤ 1 >

-$\mathcal{Abd}$:

¬ abd(rapidAddAbdOfarm). ¬ abd(inExCycloOfUpperarm).
¬ abducible(microFlexExtOfElbow). ¬ abd(pronosupinationOfForearm).
¬ abd(fixUpperarm). ¬ abd(activeLeftUpperarmMsclsStrong).

40 new characteristic clause including

¬ connected(vibrato, pronosupinationOfForearm) ∨
¬ abd(inExCycloOfUpperarm) ∨ ¬ abd(pronosupinationOfForearm).

Related Work

- Theorist (Poole, 1988) – rule abducibles (strong bias)
- **metabolic graphs** in *Robot Scientist* (Reiser et al., 2001) – complicated style of "AND" handling:

G = (V,E)

$$\left\{\begin{array}{l} edge(X,Y) \leftarrow reaction(A,B),\ A \subseteq X,\ Y = X \cup B. \\ path(X,Y) \leftarrow edge(X,Y). \\ path(X,Y) \leftarrow edge(X,Z),\ path(Z,Y). \end{array}\right.$$

- (Ray & Inoue, 2007) – no "AND" handling
- CF-induction (Yamamoto, Inoue & Doncescu, 2009) – predicate invention is only realized by inverse resolution

Summary

- Simple and powerful method for **rule abduction**.
- *Multiple observations* are explained at once.
- Allows *full clausal theories* for background knowledge.
- “AND” connective can be dealt with disjunction.
- *Empirical rules* are explained by *hidden rules*.
- Predicate invention is realized as **existentially quantified hypotheses**.
- Induction is realized by **meta-level abduction** – SOLAR as an inductive inference engine
- Application to **skill science** as well as **systems biology** – Future work: cancer diagnosis & therapy.