渕一博記念コロキウム
『論理と推論技術：四半世紀の展開』

非単調性と帰納論理を取り入れたことで論理プログラミングはどう変わったか？

国立情報学研究所
情報学プリンシプル研究系

井上 克巳

2007年10月20日

情報・システム研究機構
国立情報学研究所
National Institute of Informatics

Incorporating Nonmonotonic and Inductive Inference into Logic Programming

October 20, 2007

Katsumi Inoue

National Institute of Informatics

Thanks to:

Andrei Doncescu, Koji Iwanuma, Hidetomo Nabeshima, Oliver Ray, Chiaki Sakama, Yoshitaka Yamamoto

History of Thinking Machines

1936. Turing Machine
1943. McCulloch & Pitts: Boolean circuit model of brain
1950. Turing: “Computing Machinery and Intelligence”
1956. Dartmouth meeting: “Artificial Intelligence” adopted
1958. McCarthy: “Programs with Common Sense”
1950s. Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist
1960. McCarthy: LISP
1965. Robinson: complete algorithm for logical reasoning
1969. McCarthy & Hayes: situation calculus, frame problem
1971. Colmerauer & Kowalski: Prolog
1970s. Early development of knowledge-based systems
1980. McCarthy: circumscription
1982. **Japan's Fifth Generation Computer Project**
1986. Neural networks return to popularity
1995. Emergence of intelligent agents
1997. Deep Blue

History of Logic Programming

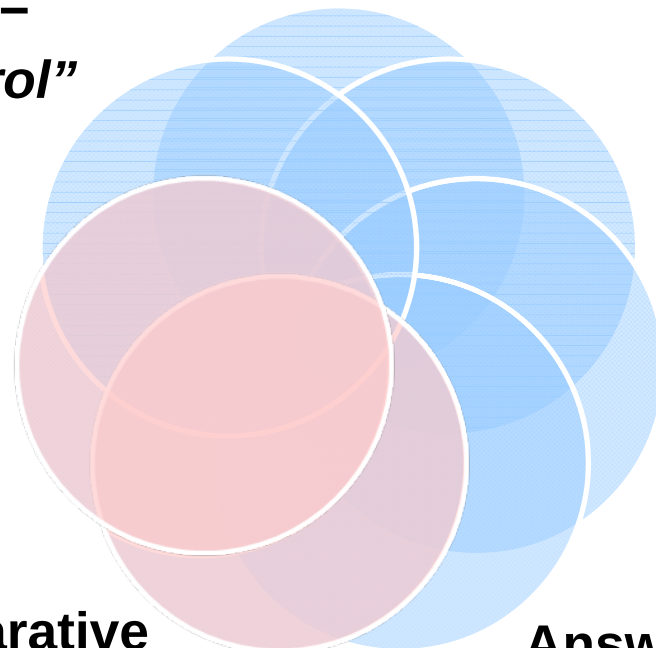
1960. Davis & Putnum: testing satisfiability of propositional formulas
1962. Davis, Logemann & Loveland: improvement of DP → SAT
1965. Robinson: resolution principle
1968. Loveland: Model Elimination
1971. Kowalski & Kuehner: SL-resolution
1972. Colmerauer & Roussel: "*Programmation en Logique*"
1972. Kowalski: "The predicate calculus as a programming language"
1973. Hill: LUSH (SLD-)resolution
1974. van Emden & Kowalski: Scott's fixpoint & Tarskian semantics
1977. Warren (David): Edinburgh Prolog compiler
1978. Clark: negation as failure (predicate completion)
1982. **Japan's Fifth Generation Computer Project**
1980s. Shapiro, Clark, Ueda: Concurrent Logic Programming
1987. Jaffer & Lassez: Constraint Logic Programming
1988. Gelfond & Lifschitz: stable model semantics
1991. Muggleton: "Inductive Logic Programming"
1992. Kakas, Kowalski & Toni: "Abductive Logic Programming"
1990s. Poole, Sato: Probabilistic Logic Programming

Interpreting Horn Logic Programs

$$H \ :- \ B_1, \ \ldots, \ B_n.$$

"Algorithm = Logic + Control"
(Kowalski)

Procedural reading
(Hewitt, Colmerauer, Hayes)



Logical formula / Knowledge representation
(McCarthy)

Theorem proving
(Loveland, Kowalski, Hill)

Declarative semantics
(van Emden & Kowalski)

Answer extraction
(Slagle, Green, Colmerauer)

Extending Horn Logic Programs

$$H_1; \ldots; H_m; \textbf{not } H_{m+1}; \ldots; \textbf{not } H_n \leftarrow B_1, \ldots, B_m, \textbf{not } B_{m+1}, \ldots, \textbf{not } B_n$$

- Definite (Horn) program (H, B: atom)
- Normal logic program
- Extended logic program (H, B: literal)
- Extended disjunctive program
- General extended disjunctive program
- Nested program (H, B: rule)

Logic Programming and Nonmonotonic Reasoning

1969. McCarthy & Hayes: frame problem
1978. Reiter: closed-world assumption
1979. Clark: negation as failure (Compl)
1980. McCarthy: circumscription (Circ)
1980. Reiter: default logic (DL)
1982. Reiter: Circ implies Compl (sometimes)
1985. Moore: autoepistemic logic (AEL)
1987. Gelfond & Lifschitz: stratified LP as prioritized Circ
1988. Gelfond & Lifschitz: stable model semantics (LP as AEL)
1990. Gelfond & Lifschitz: answer set semantics (LP as DL)
1992. Inoue, Koshimura & Hasegawa: answer set computation in KL1
1996. Niemelä & Simons: smodels
1997. Eiter, Faber, Leone & Pfeifer: DLV
2007. Lifschitz, Lin: 2^{nd}-order formalization of stable models (LP as Circ)

Answer Set Programming

A **program** is regarded as the ***constraints*** to be satisfied by **solutions**. Each solution is obtained by computing an ***answer set*** **(stable model)** of the program.
A program may have no, one, or multiple answer sets.

Program:

p ; ***not*** $p \leftarrow$,
q ; ***not*** $q \leftarrow$,
r ; ***not*** $r \leftarrow$, } **generator**

$s \leftarrow p, r$,
$s \leftarrow p, q,$ ***not*** r , } **tester**

$\leftarrow$ ***not*** s. } **goal**

Answer Sets: $\{p, q, r, s\}$, $\{p, r, s\}$, $\{p, q, s\}$.

Answer Set Programming

A **program** is regarded as the ***constraints*** to be satisfied by **solutions**. Each solution is obtained by computing an ***answer set*** **(stable model)** of the program.
A program may have no, one, or multiple answer sets.

Program:

$$p \leftarrow \textbf{not}\ q, \textbf{not}\ r,$$
$$q \leftarrow \textbf{not}\ p, \textbf{not}\ r,$$
$$r \leftarrow \textbf{not}\ p, \textbf{not}\ q,$$

generator (exclusive choice)

$$s \leftarrow p, r,$$
$$s \leftarrow q, \textbf{not}\ r,$$

tester

$$\leftarrow \textbf{not}\ s.$$

goal

Answer Sets: $\{q, s\}$.

Nonmonotonic Reasoning in ICOT

1984 Kitakami, Kunifuji, Miyachi & Furukawa: knowledge assimilation (SLP'84)

1986 Kunifuji, Tsurumaki & Furukawa: hypothesis-based reasoning

1986 Goebel, Poole & Furukawa: theory formation (ICLP'86)

1988 Ishizuka: *Hypothetical Reasoning WG*

1988 Arima, Satoh: work on circumscription (FGCS'88)

1989 Sakama: possible model semantics for disjunctive programs

1989 *Nicolas Helft & David Poole in ICOT*

1990 Inoue & Helft: circumscriptive theorem prover

1991 Helft, Inoue & Poole: QA in circumscription (IJCAI-91)

1991 *Mark Stickel in ICOT*

1991 Inoue: SOL-resolution (IJCAI-91)

1991 Inoue, Satoh & Iwayama: abduction in answer set semantics (ICLP'91)

1992 Inoue, Koshimura & Hasegawa: answer set comp. on MGTP (CADE'92)

1992 Ohta & Inoue: parallel abductive reasoning on MGTP & ATMS (FGCS'92)

1993 Inoue, Ohta, Hasegawa & Nakashima: abduction on MGTP (IJCAI-93)

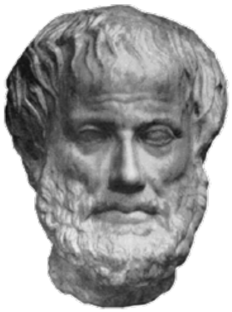
1993 Inoue & Sakama: fixpoint semantics for abductive programs (ICLP'93)

Inoue & Sakama or Sakama & Inoue

1993.	ICLP’93	ICLP’93	LPNMR’93				
1994.		JAR	KR’94		ICLP’94		
1995.		ICLP’95		IJCAI-95		LPNMR’95	JLC
1996.	JLP	JICSLP’96					
1997.							
1998.	KR’98		JLP				
1999.	AMAI		LPNMR’99			IJCAI-99	
2000.		AIJ			JLP		
2001.							
2002.	ICLP’02						
2003.			TPLP				
2004.	JELIA’04						
2005.	CLIMA-05	IJCAI-05	ILP’05				
2006.		LJ.IGPL	CLIMA-06		ICLP’06	LOPSTR’06	
2007.		AAMAS’07			AAAI-07		
2008.	ACM.TOCL						

Abduction, Induction and Deduction

Analytic Reasoning



Deduction (LP)

consequence:
from *prior knowledge*
to *necessary implications*

Synthetic Reasoning

Induction (ILP)

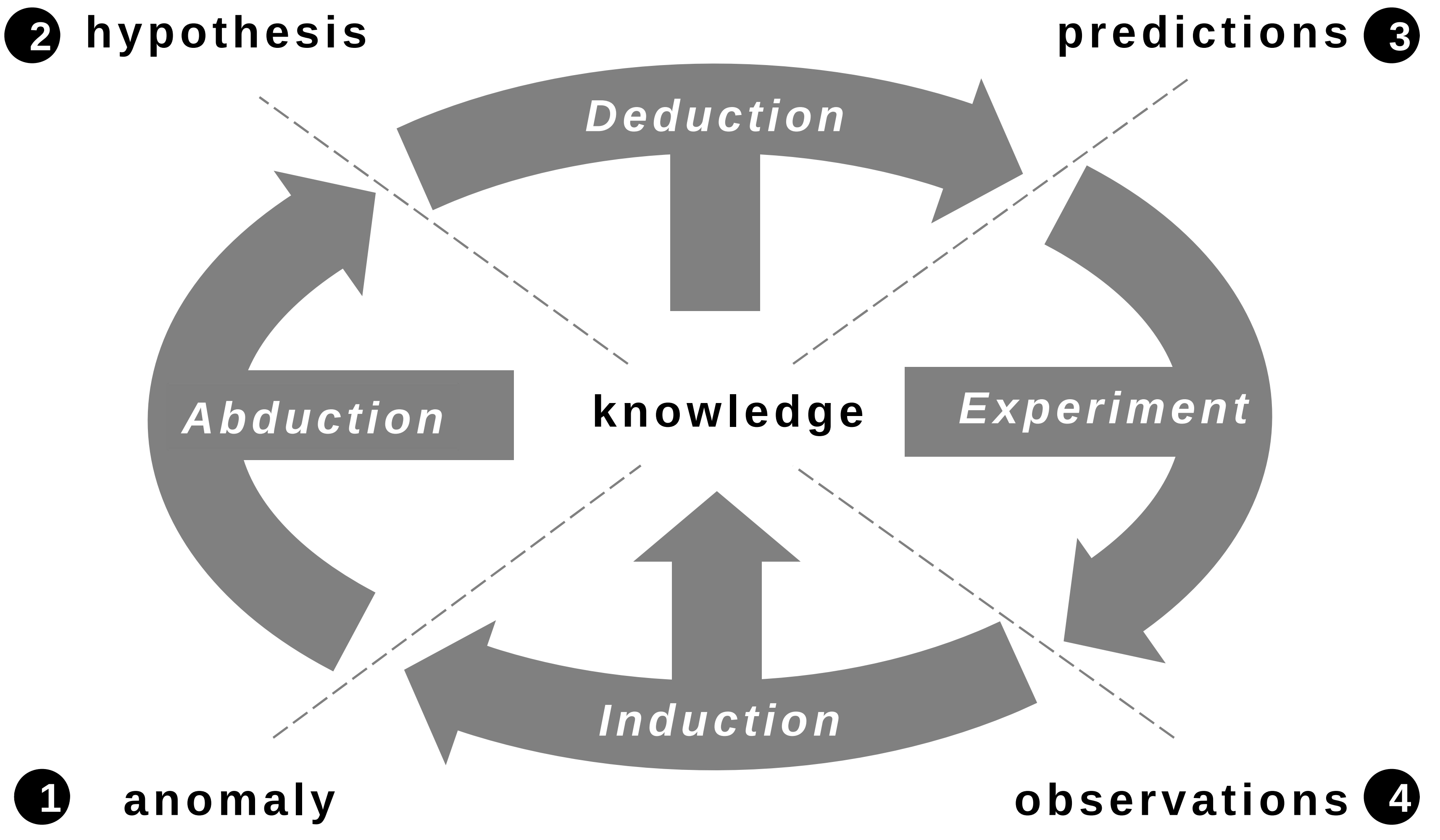
generalization:
from *observed samples*
to *wider populations*

Abduction (ALP)

explanation:
from *given effects*
to *possible causes*

Scientific Discovery

Scientific Knowledge Development



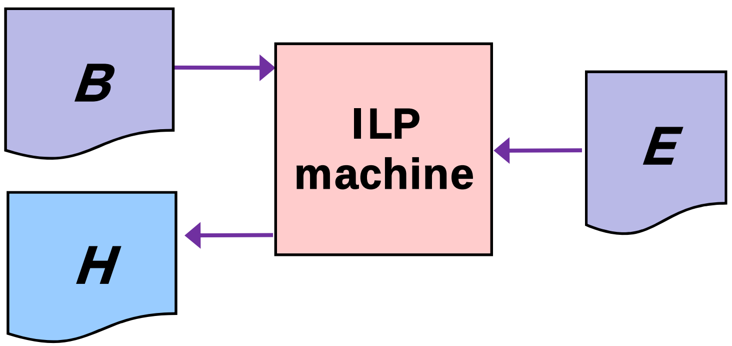
Abduction and Induction: Logical Framework

Input:

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

Output:

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



Inverse Entailment (IE)

Computing a hypothesis H can be done deductively by:

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

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

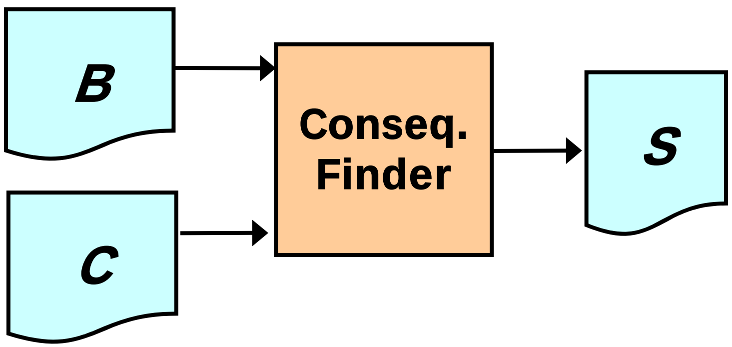
Consequence Finding

Input:

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

Output:

- S : the (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, $C = \neg E$ and $S = \neg H$: $\boldsymbol{B \wedge \neg E \models \neg H}$.

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:

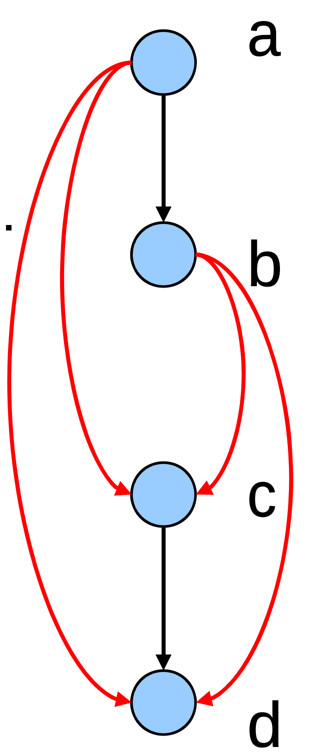
¬path(a,d).

Production field:

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

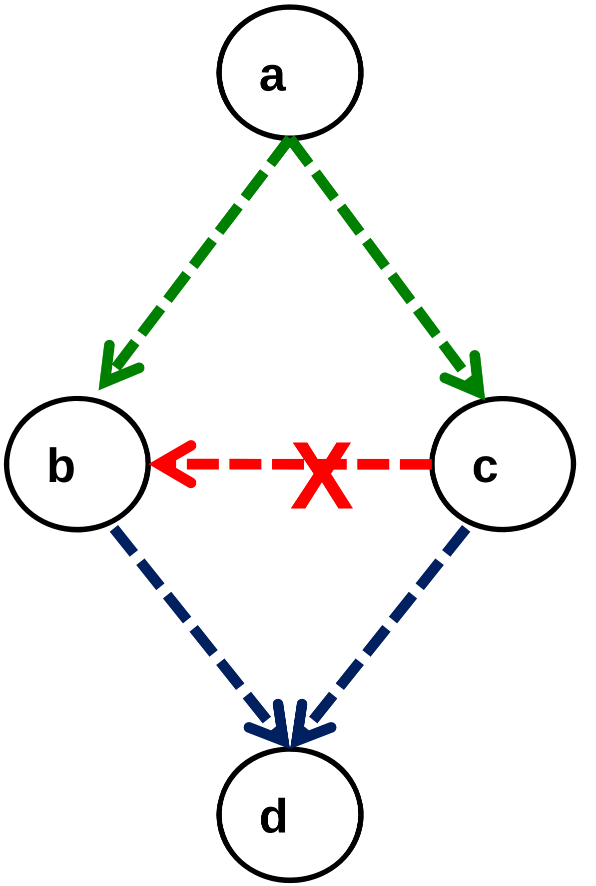
Output of SOLAR:

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



Metabolic Pathway (Ray & Inoue, DS'07)

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



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

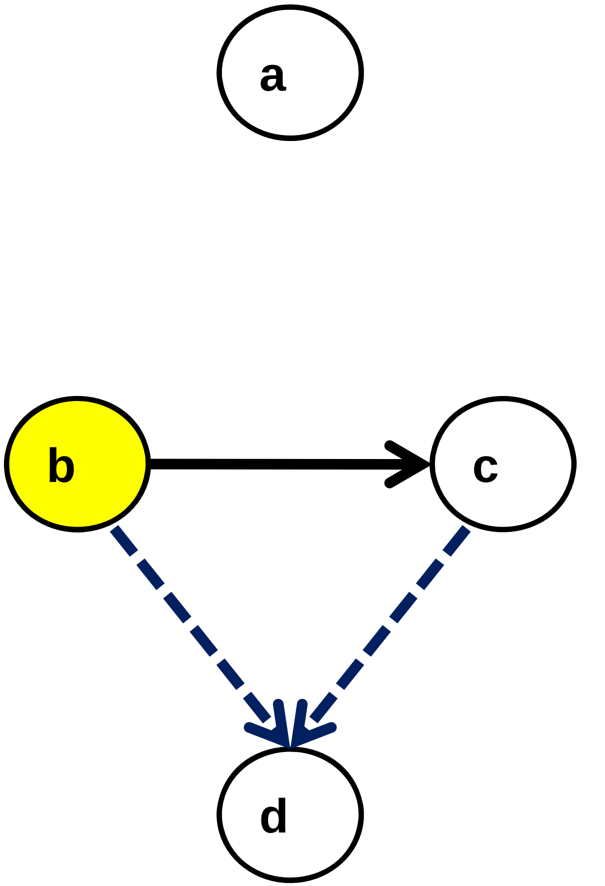
% (from which U) is there a path to d?

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

% assuming reactions from some V to W

Metabolic Pathway: A Solution

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



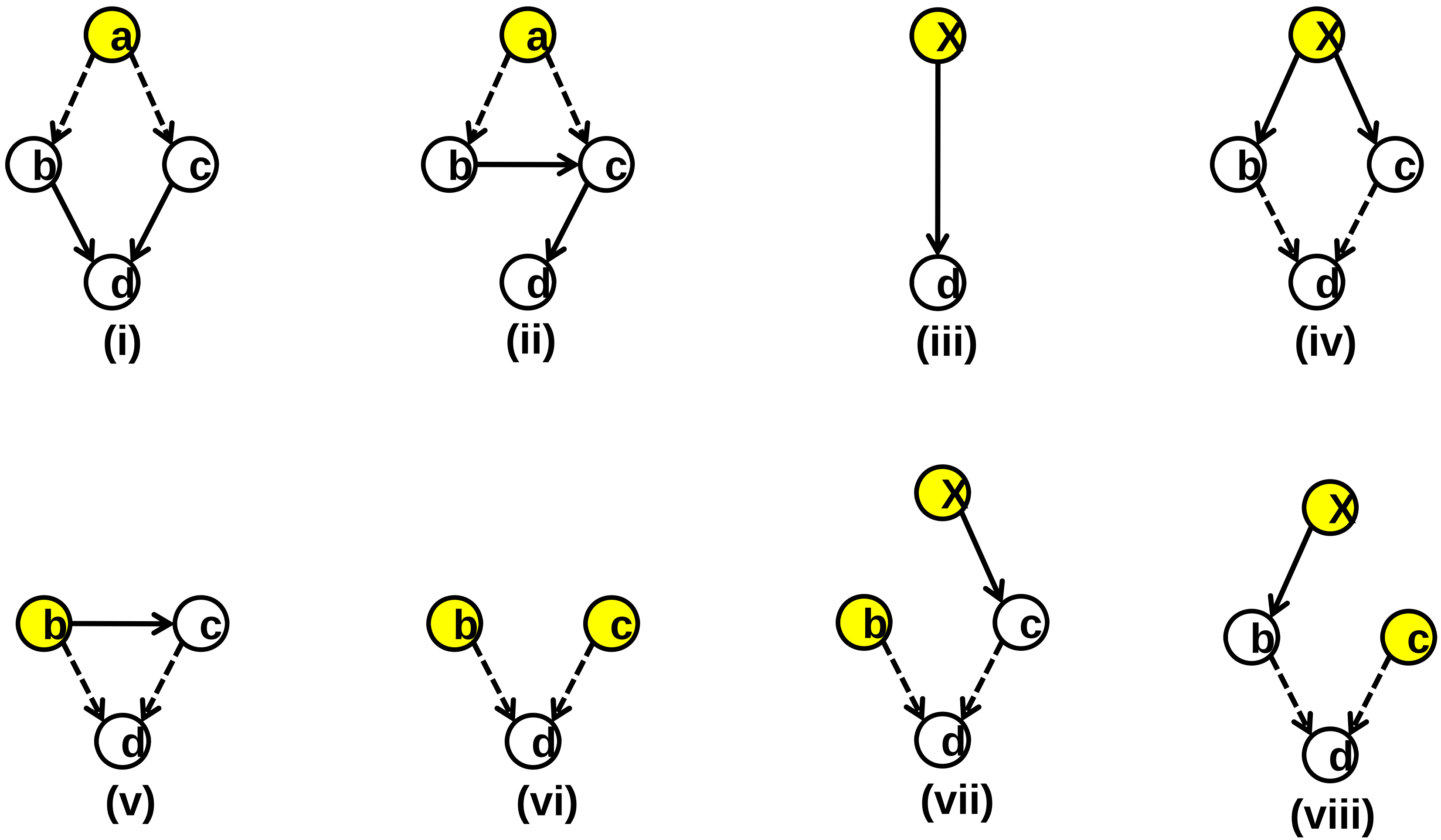
$$E\sigma = \{pathway(b,d)\}$$

% 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 (Ray & Inoue, DS'07)

given

$\boldsymbol{B}$	**theory (background knowledge)**
$\boldsymbol{E}$	**goal (set of given observations)**
$\boldsymbol{A}$	**abducibles (set of possible assumptions)**

find

$\boldsymbol{H \subseteq A}$	**explanation (set of assumptions)**
$\boldsymbol{\Theta}$	**answer (SET of variable bindings)**

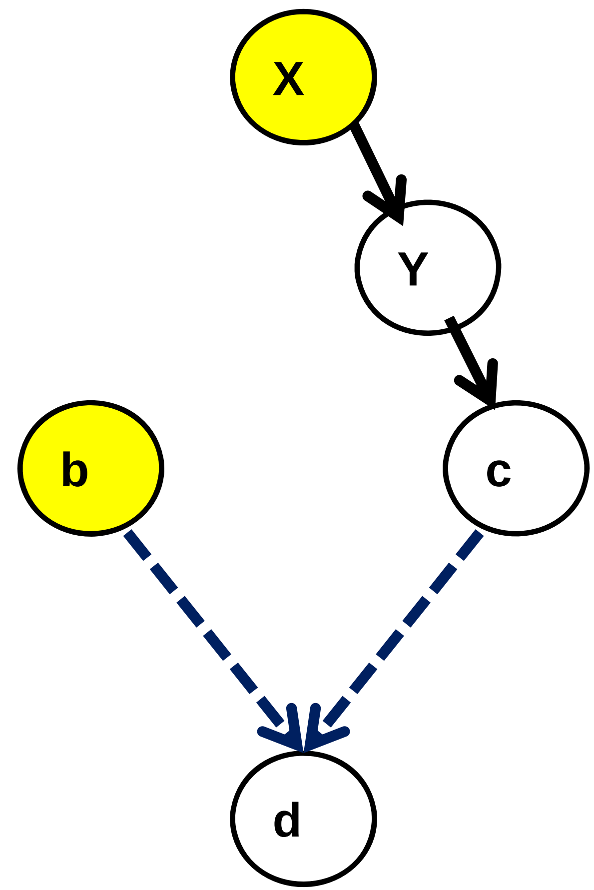
where

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

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

Metabolic Pathway: Another Solution

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



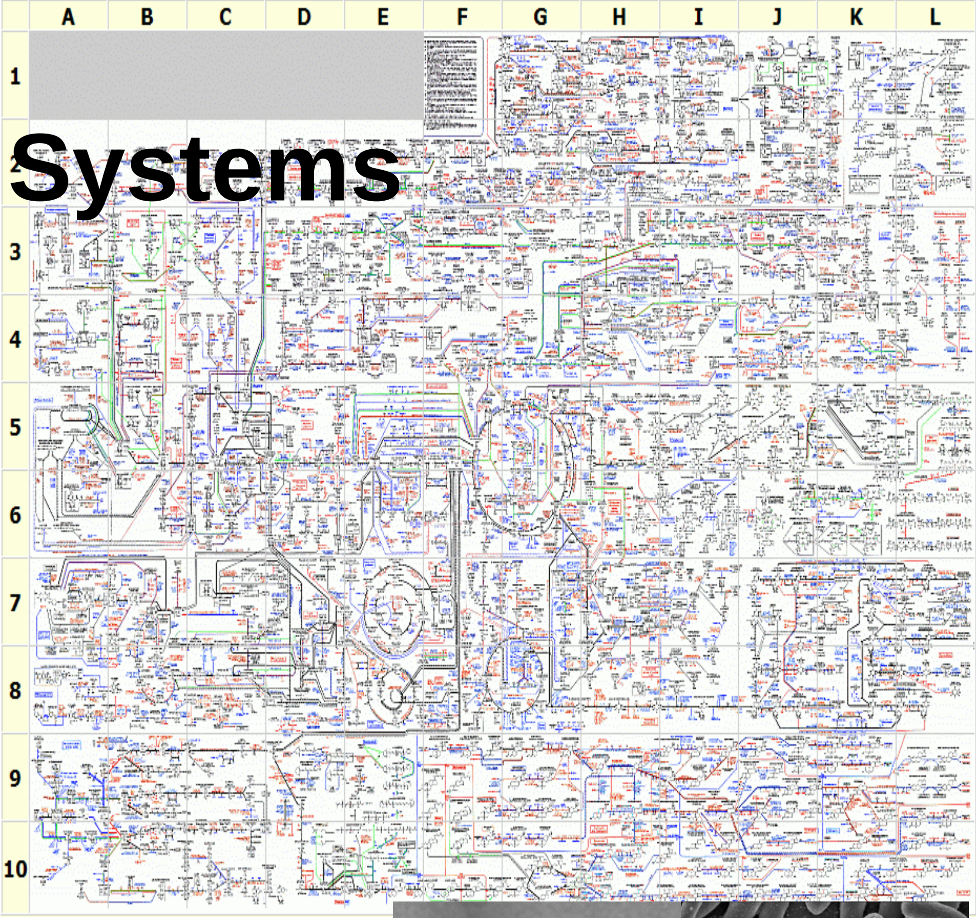
$$\Theta = \{\{U/b\},\{U/X\}\}$$

% there is a path from b or X to d

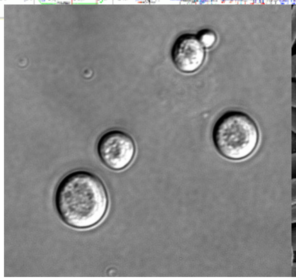
$$H = \{reaction(X,Y), reaction(Y,c)\}$$

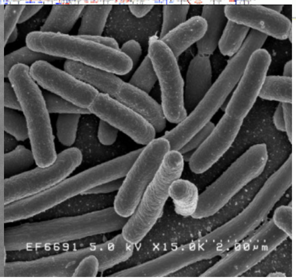
% assuming reactions from X to Y and from Y to c

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**.

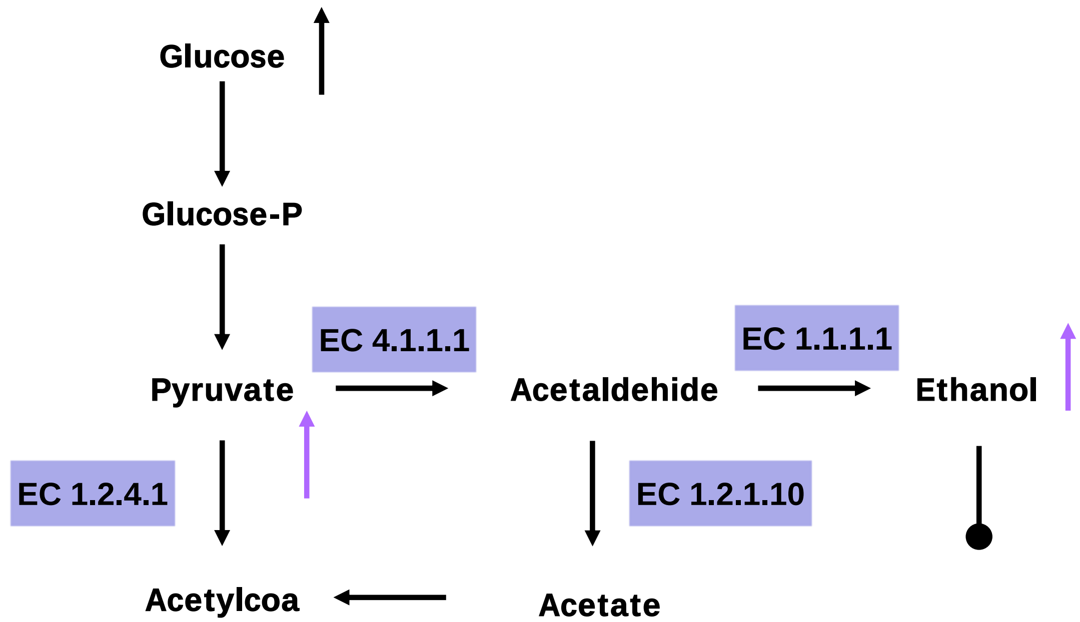
Inverse Entailment for Induction

- **Horn clauses** for background theories
  - **Progol** (Muggleton, 1995)

- **Full clausal theories** (non-Horn clauses) for background, example, and hypothesis theories.
  - **CF-induction** (Inoue, ILP-2001; Yamamoto, Ray & Inoue, LLLL-2007)
  - **fc-HAIL** (Ray & Inoue, ILP-2007)

- Note: CF-induction is the only existing ILP system that is **complete** for full clausal theories.

Example: Metabolic Pathway (Pyruvate)



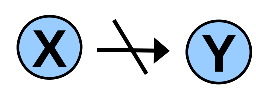
- ***B***:

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

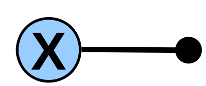
concentration(glucose, up).

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).

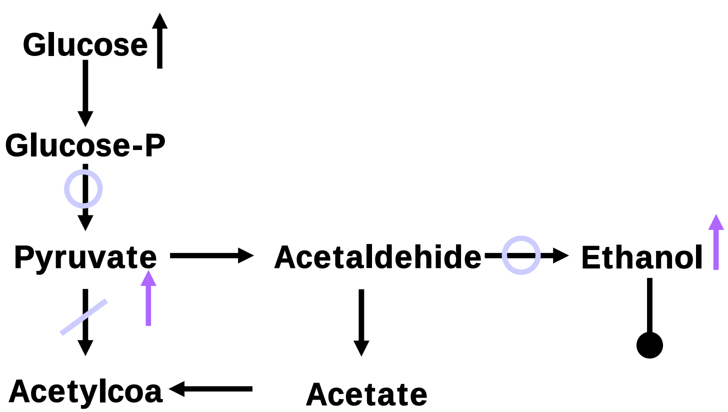
Example: Outputs of CF-induction

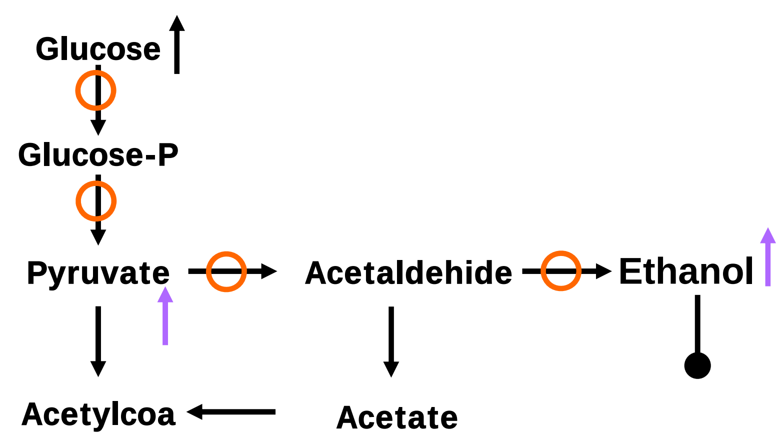
- *H_1:*

¬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).





Conclusion

- Algorithm = Logic + Control.
- Logic programming is still alive.
- Is this GOFAI?
- Dealing with **incomplete knowledge** and **elaboration tolerance** — default reasoning, planning, prediction, explanation, hypothesis formation, model generation.
- Inference-based hypothesis-finding in biochemical networks.
- Logic is a glue to combine mathematical/abstract models with the real world.
- Get back to logic (programming)!