On Computing all Abductive Explanations from a Propositional Horn Theory

Kaz Makino
(Graduate School of Engineering Science, Osaka Univ.)

Joint work with Thomas Eiter
(Technische Universität Wien)

Outline

1. 3 reasoning mechanisms
2. Abduction from Horn theories
3. Generating abductive explanations from Horn theories
4. Model-based representation for Horn theories

3 Reasoning Mechanisms

Deduction: fact $\cup$ knowledge base $\models$?

Induction: fact $\cup$? $\models$ observation

Abduction: ? $\cup$ knowledge base $\models$ observation

Deduction

Fact: battery is down

knowledge: if the battery is down, the car will not start

……………………………

$\Rightarrow$ The car will not start

3 Reasoning Mechanisms

Deduction: fact $\cup$ knowledge base $\models$?

Induction: fact $\cup$? $\models$ observation

Abduction: ? $\cup$ knowledge base $\models$ observation

Induction

$\left\{\begin{array}{l}\text{Fact: battery is down} \\ \text{Observation: The car will not start}\end{array}\right.$

$\Rightarrow$ Rule: if the battery is down, the car will not start

3 Reasoning Mechanisms

Deduction: fact $\cup$ knowledge base $\models$?

Induction: fact $\cup$? $\models$ observation

Abduction: ? $\cup$ knowledge base $\models$ observation

Abduction

knowledge: if the battery is down, the car will not start

…………………………………

Observation: The car will not start

$\Rightarrow$ Fact: battery is down

Abduction (formulated by C.S. Peirce ’31-’58)

Widely used in Computer Science and AI

Basis for

- Truth Maintenance Systems (TMS, ATMS)
- Clause management Systems (CMS)
- Diagnosis
- Database Update, ...

Propositional Horn Knowledge Base

Propositional variables: $x_1, x_2, \cdots, x_n \in \{0,1\}$

Knowledge $f : \{0,1\}^n \to \{0,1\}$

CNF (conjunctive normal form): E.g.,

$$\varphi = (\overline{x}_2 \vee \overline{x}_3 \vee x_4)(\overline{x}_3 \vee \overline{x}_4 \vee x_2)(\overline{x}_2 \vee x_1)$$

Horn CNF: at most 1 positive literal in each clause

Propositional Horn Knowledge Base

Horn CNF: at most 1 positive literal in each clause

$$\varphi = (\overline{x}_2 \vee \overline{x}_3 \vee x_4)(\overline{x}_3 \vee \overline{x}_4 \vee x_2)(\overline{x}_2 \vee x_1)$$

Horn clause: $(\overline{x}_{i_1} \vee \overline{x}_{i_2} \vee \cdots \vee \overline{x}_{i_k} \vee x_{i_0})$

$\Updownarrow$

Horn rule: $x_{i_1} \wedge x_{i_2} \wedge \cdots \wedge x_{i_k} \to x_{i_0}$

(antecedent/consequent: may be empty)

$$\varphi^* = (x_2 x_3 \to x_4)(x_3 x_4 \to x_2)(x_2 \to x_1)$$

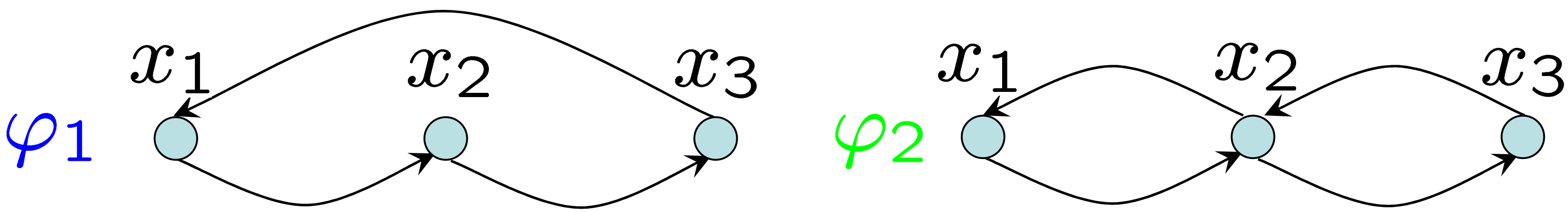
Core language in AI and logic programming

Horn CNF representation is not unique

E.g.,

$$\varphi_1 = (x_1 \to x_2)(x_2 \to x_3)(x_3 \to x_1)$$

$$\varphi_2 = (x_1 \to x_2)(x_2 \to x_1)(x_2 \to x_3)(x_3 \to x_2)$$



$$mod(\varphi_1) = mod(\varphi_2) = \{(111), (000)\}$$

$$||$$

$$\{v \in \{0,1\}^n \mid \varphi_1(v) = 1\}$$

Explanations

φ: a Horn CNF

q : a propositional variable

An explanation for q from φ: a minimal set E s.t.

(1) $\varphi \cup E \models q$

(2) $\varphi \cup E$ is satisfiable

$$\varphi \cup E \equiv \varphi \wedge \bigwedge_{x \in E} x$$

$\varphi \models \psi$: $\varphi(v) = 1$ implies $\psi(v) = 1$ for all $v \in \{0,1\}^n$

(1) $\varphi \models (\bigwedge_{x \in E} x \rightarrow q)$

(2) $\exists v \in mod(\varphi)$ s.t. $(\bigwedge_{x \in E} x)(v) = 1$

Explanations

φ : a Horn CNF

q : a propositional variable

An explanation for q from φ : a minimal set E s.t.

(1) $\varphi \cup E \models q$

(2) $\varphi \cup E$ is satisfiable

Abduction: $\underset{E}{?} \cup \underset{\varphi}{\text{knowledge base}} \models \underset{q}{\text{observation}}$

E.g., Explanations for x_1 from

$$\varphi = (x_2 x_3 \to x_4)(x_3 x_4 \to x_2)(x_2 x_3 \to x_1)$$

$\varphi \cup \{x_1\} \models x_1$

$\varphi \cup \{x_2, x_3\} \models x_1$

$\varphi \cup \{x_3, x_4\} \models x_1$

(1) $\varphi \cup E \models x_1$
(2) $\varphi \cup E$ is satisfiable

$$E = \{x_1\}, \{x_2, x_3\}, \{x_3, x_4\}$$

$E = \{x_1\}$: trivial explanation for x_1

Well-known:

Finding a nontrivial explanation E is poly. time.
Finding an explanation $E \subseteq A$ is NP-hard.
(Selman & Levesque '90)

Abduction: $\underset{E}{\textcolor{red}{?}} \cup \underset{\varphi}{\text{knowledge base}} \models \underset{q}{\text{observation}}$

Car diagnosis

φ: if the battery is down, the car will not start
if the gas tank is empty, the car will not start
………………………………
q: The car will not start

E: {battery is down}, {gas tank is empty}, …

1. Generate all possible explanations
2. Find a real one from them

Can we generate all (poly. many) explanations efficiently ?

Conjecture by Selman & Levesque ('90)

Generating $O(n)$ explanations is NP-hard, even if there are only few explanations overall.

Eiter & Makino (2002) disproved it

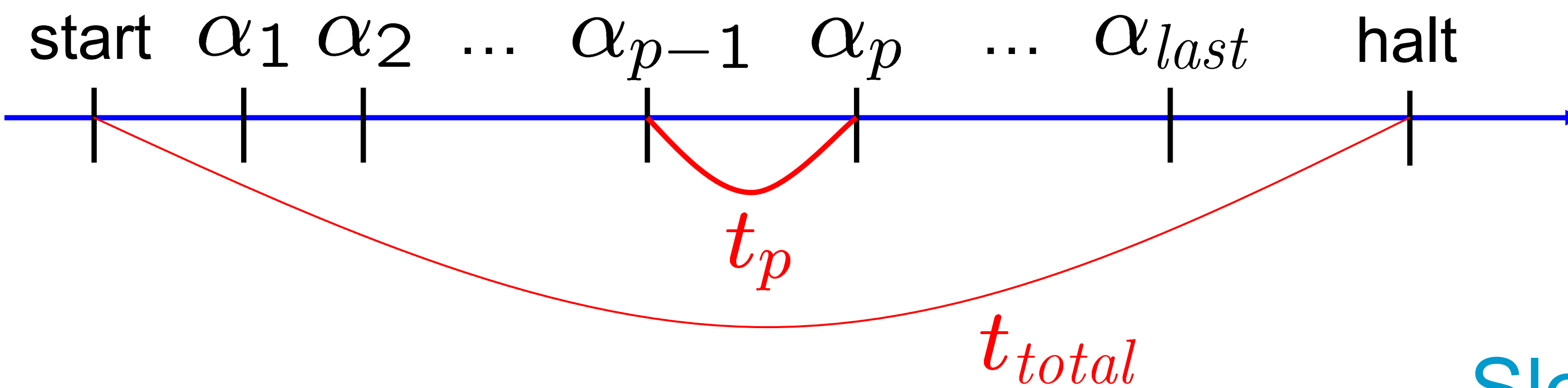
Note: Exponentially many explanations might exist.

$$\varphi = \bigwedge_{i=1}^{n} (x_i \rightarrow y_i) \wedge (y_1 y_2 \ldots y_n \rightarrow q)$$

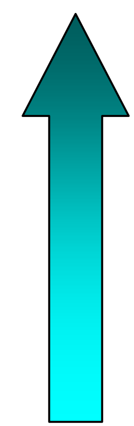
2^n explanations

$$\mathcal{E} = \{\{e_1, e_2, \ldots, e_n\} \mid e_i \in \{x_i, y_i\}\}$$

Complexity of generating problem



Slow



Fast

Output P : $t_{total} = poly(\mathrm{input} + \mathrm{output})$

Incremental P : $t_p = poly(\mathrm{input} + \sum_{i=1}^{p-1} |\alpha_i|)$

P delay : $t_p = poly(\mathrm{Input})$

Prime implicate c of f

$$f \models c,\ f \not\models c' \text{ for any } c' \subsetneq c$$

Ex. $f = (x_1 \to x_2)(x_2 \to x_3)(x_3 \to x_1)$

$$(mod(f) \ = \{(111), (000)\})$$

- $\overline{x}_1 \vee x_2 \, (\equiv x_1 \to x_2)$

  $$f \models \overline{x}_1 \vee x_2,\ f \not\models \overline{x}_1,\ f \not\models x_2.$$

- $\overline{x}_2 \vee x_1 \, (\equiv x_2 \to x_1)$

  $$f \models \overline{x}_2 \vee x_1,\ f \not\models \overline{x}_2,\ f \not\models x_1.$$

Explanations and prime implicates

Explanation: minimal E s.t.

(1) $\varphi \cup E \models q \Leftrightarrow \varphi \models (\bigwedge_{x \in E} x \rightarrow q)$

(2) $\varphi \cup E$ is satisfiable

nontrivial explanation for q

$\Updownarrow$

prime implicate containing q

How to generate all prime implicates

$$c_1 = \left(\bigvee_{x \in P(c_1)} x \vee \bigvee_{x \in N(c_1)} \overline{x}\right)$$

$$c_2 = \left(\bigvee_{x \in P(c_2)} x \vee \bigvee_{x \in N(c_2)} \overline{x}\right)$$

$$(P(c_1) \cap N(c_2) = \{z\}, N(c_1) \cap P(c_2) = \emptyset)$$

resolvent

$$c_3 = \left(\bigvee_{x \in (P(c_1) \setminus \{z\}) \cup P(c_2)} x \vee \bigvee_{x \in N(c_1) \cup (N(c_2) \setminus \{z\})} \overline{x}\right)$$

E.g., $(x_2 \vee \overline{x}_3 \vee \overline{x}_4)$: resolvent of

$(x_1 \vee \overline{x}_3 \vee \overline{x}_4)$ & $(\overline{x}_1 \vee x_2 \vee \overline{x}_3)$

$$\boxed{f \models c_1, c_2 \Longrightarrow f \models c_3}$$

Procedure Resolution

Input: A CNF $\varphi = \bigwedge_{i=1}^{m} c_i$ representing f.

Output: All prime implicates of f.

Step 1: $S := \{c_i \mid i = 1, 2, \cdots, m\}$.

Step 2: Repeat (s) simplification and (r) resolution.

(s) Remove c from S if $\exists c^* \in S$ s.t. $c^* \models c$. $c^* \subsetneq c$

(r) Add a resolvent of two clauses in S.

Step 3: Output all clauses in S.

Ex. $\varphi = (\overline{x}_1 \vee \overline{x}_4)(\overline{x}_4 \vee \overline{x}_3)(\overline{x}_1 \vee x_2)(\overline{x}_3 \vee \overline{x}_5 \vee x_1)$

$S_0 = \{\overline{x}_1 \vee \overline{x}_4, \overline{x}_4 \vee \overline{x}_3, \overline{x}_1 \vee x_2, \overline{x}_3 \vee \overline{x}_5 \vee x_1\}$

$S_1 = S_0 \cup \{\cancel{\underline{\overline{x}_3 \vee \overline{x}_5 \vee \overline{x}_4}}, \ \underline{\overline{x}_3 \vee \overline{x}_5 \vee x_2}\}$

$S_2 = S_1 \setminus \{\overline{x}_3 \vee \overline{x}_5 \vee \overline{x}_4\}$

Prop. [Blake ('37), Brown ('68), Quine ('55), Samson-mills ('54)]
Resolution procedure generates all prime implicates.

Prop. Even if φ is Horn, resolution procedure may require exponential time.

Prop. There is no output P algorithm for generating all prime implicates, unless P=NP.

Procedure Resolution

Input: A CNF $\varphi = \bigwedge_{i=1}^{m} c_i$ representing f.

Output: All prime implicates of f.

Step 1: $S := \{c_i \mid i = 1, 2, \cdots, m\}$.

Step 2: Repeat (s) simplification and (r) resolution.

(s) Remove c from S if $\exists c^* \in S$ s.t. $c^* \models c$.

(r) Add a resolvent c_3 of two clauses c_1, c_2 in S.

Step 3: Output all clauses in S.

Modification

(1) Input resolution: $c_1 \in \varphi$

(2) Add a prime implicate c' s.t. $c' \models c_3$

(3) Output c' in (r) immediately, if c' is new.

Th. [Boros, Crama, Hammer ('90)]

If φ is Horn, then input-resolution procedure generates all prime implicates in incremental P time.

nontrivial explanation for q

$$\Updownarrow$$

prime implicate containing q

Procedure Resolution

Input: A CNF $\varphi = \bigwedge_{i=1}^{m} c_i$ representing f.

Output: All prime implicates of f.

Step 1: $S := \{c_i \mid i = 1, 2, \cdots, m\}$.

Step 2: Repeat (s) simplification and (r) resolution.

(s) Remove c from S if $\exists c^* \in S$ s.t. $c^* \models c$.

(r) Add a resolvent c_3 of two clauses c_1, c_2 in S.

Step 3: Output all clauses in S.

Modification

(1) Input resolution: $c_1 \in \varphi$

(2) Add a prime implicate c' s.t. $c' \models c_3$

(3) Output c' in (r) immediately, if c' is new.

(4) $c_2 \ni q$

Th. [Eiter, Makino (2002, 2003)]
All explanations for q from a Horn CNF can be computed with P delay.

Sketch:

Cor. P many explanations for q from a Horn CNF can be computed in (input) P time.

Explanations for a negative literal

nontrivial explanation for a negative literal $\overline{q}$

$\Updownarrow$

prime implicate containing $\overline{q}$

Procedure Resolution

Input: A CNF $\varphi = \bigwedge_{i=1}^{m} c_i$ representing f.

Output: All prime implicates of f.

Step 1: $S := \{c_i \mid i = 1, 2, \cdots, m\}$.

Step 2: Repeat (s) simplification and (r) resolution.

(s) Remove c from S if $\exists c^* \in S$ s.t. $c^* \models c$.

(r) Add a resolvent c_3 of two clauses c_1, c_2 in S.

Step 3: Output all clauses in S.

Modification

(1) Input resolution: $c_1 \in \varphi$

(2) Add a prime implicate c' s.t. $c' \models c_3$

(3) Output c' in (r) immediately, if c' is new.

(4) $\cancel{c_2 \ni q}$ $\Rightarrow$ $c_2 \ni \overline{q}$

Explanations for a negative literal

Prop. [Eiter, Makino (2003)]
Our resolution procedure does not generate all explanations for $\overline{q}$ from a Horn CNF.

Th. [Eiter, Makino (2003)]
There exists no output P algorithm for generating all explanations for $\overline{q}$ from a Horn CNF, unless P=NP.

Th. [Eiter, Makino (2003)]
All explanations for $\overline{q}$ from an acyclic Horn CNF can be computed in incremental P time.

Summary

Knowlege	Explanations E w.r.t. $A = Lit$	
	query q	query $\overline{q}$
Horn CNF	P delay	no output P
Acyclic Horn CNF	P delay	incremental P
Characteristic set	MDual	MDual

Knowlege	Explanations E w.r.t. $A \subseteq Lit$	
	query q	query $\overline{q}$
Horn CNF	coNPc	coNPc
Acyclic Horn CNF	coNPc	coNPc
Characteristic set	MDual	MDual

Model-based reasoning

Deduction $\varphi \models \psi$

$$\Updownarrow$$

$\varphi(v) = 1$ implies $\psi(v) = 1$

E.g., $\varphi = (\overline{x}_2 \vee \overline{x}_3 \vee x_4)(\overline{x}_3 \vee \overline{x}_4 \vee x_2)(\overline{x}_2 \vee \overline{x}_3 \vee x_1)$

$$mod(\varphi) = \{(0000), (0001), (0010), (0100), (0101), (1000), (1001), (1010), (1100), (1101), (1111)\}$$

All models $v \in mod(\varphi)$ satisfy $(x_3 x_4 \rightarrow x_1)$

$$\Longrightarrow \quad \varphi \models (x_3 x_4 \rightarrow x_1)$$

$(0101) \in mod(\varphi)$ does not satisfy $(x_2 x_4 \rightarrow x_1)$

$$\Longrightarrow \quad \varphi \not\models (x_2 x_4 \rightarrow x_1)$$

$|mod(\varphi)|$: large $\Longrightarrow$ inefficient

[P. N. Johnson-Laird ('83)].

Humans typically argue by just looking at some examples.

Some models $v \in mod(\varphi)$ satisfy $\psi(v) = 1$

conclude $\varphi \models \psi$; otherwise, $\varphi \not\models \psi$

Of course, incorrect !

$\exists S (\subseteq mod(\varphi))$ s.t.

$\forall v \in S$: $\psi(v) = 1$

$\Updownarrow$

$\forall v \in mod(\varphi)$: $\psi(v) = 1$ (i.e., $\varphi \models \psi$)

$v \wedge w$: intersection of $v, w \in \{0,1\}^n$

$$(v \wedge w)_j = v_j \wedge w_j,\ j = 1, 2, \cdots, n$$

E.g., $v = (1100)$, $v = (0110)$

$$v \wedge w = (0100)$$

Prop. [McKinsey ('43)]

f: a Horn function

$\Longleftrightarrow$ $mod(f)$ is closed under $\wedge$

Semantic, model theoretic characterization

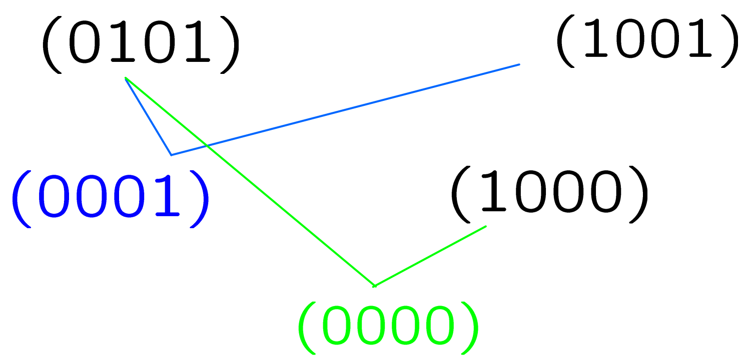
Horn CNF: syntactic characterization

Intersection closure

$$Cl_\wedge(T) = \{\bigwedge_{w \in S} w \mid \emptyset \neq S \subseteq T\}$$

E.g., $Cl_\wedge(\{(0101), (1001), (1000)\})$

$= \{(0101), (1001), (1000), (0001), (0000)\}$



Characteristic set

$$char(T) = \{v \in T \mid v \notin Cl_\wedge(T - \{v\})\}$$

$v \in char(T)$ for maximal model $v \in T$

Relational Database: generating set

Model-based representation of a Horn function

Given Characteristic set $char(T)$

Dedution: poly. time

Finding a nontrivial explanation E is poly. time.
Finding an explanation $E \subseteq A$ is poly. time.

Horn CNFs

Finding a nontrivial explanation E is poly. time.
Finding an explanation $E \subseteq A$ is NP-hard.

Summary

Knowlege	Explanations E w.r.t. $A = Lit$	
	query q	query $\overline{q}$
Horn CNF	P delay	no output P
Acyclic Horn CNF	P delay	incremental P
Characteristic set	MDual	MDual

Knowlege	Explanations E w.r.t. $A \subseteq Lit$	
	query q	query $\overline{q}$
Horn CNF	coNPc	coNPc
Acyclic Horn CNF	coNPc	coNPc
Characteristic set	MDual	MDual

Monotone Dualization

Input: A CNF φ of (a monotone function) f.

Output: Prime DNF ψ of f.

Ex. $\varphi = (x_1 \vee x_2)(x_2 \vee x_3)(x_3 \vee x_4)$

$$\equiv \cancel{x_1 x_2 x_3} \vee \cancel{x_1 x_2 x_4} \vee x_1 x_3 \vee \cancel{x_1 x_3 x_4} \vee \cdots$$

$$\equiv x_1 x_3 \vee x_2 x_3 \vee x_2 x_4 \quad = \quad \psi$$

Monotone Dualization

Input: A CNF φ of (a monotone function) f.

Output: Prime DNF ψ of f.

Many P equivalent problems

Polynomial？ OPEN

Bioch, Boros, Crama, Domingo, Eiter, Elbassioni, Fredman, Gaur, Gogic,Gottlob, Gunopulos, Gurvich, Hammer, Ibaraki, Johnson, Kameda, Kavvadias, Khachiyan, Khardon, Kogan, Krishnamurti, Lawler, Lenstra, Lovász, Mannila, Mishra, Papadimitriou, Pitt, Rinnoy Kan, Sideri, Stavropoulos, Tamaki, Toinonen, Uno, Yannakakis, ...

Polynomial？　OPEN

茨木俊秀．単調論理関数の同定問題とその複雑さ,
離散構造とアルゴリズム III,室田一雄（編）近代科学社,
pp. 1--33, 1994. Toshihide Ibaraki

Johnson. Open and closed problems in NP-completeness. Lecture given at the International School of Mathematics ``G. Stampacchia'': Summer School ``NP-Completeness: The First 20 Years'', Erice, Italy, June 20-27, 1991.

Lovasz. Combinatorial optimization: Some problems and trends, DIMACS Technical Report 92-53, 1992.

Papadimitriou. NP-completeness: A retrospective,
In: Proc. 24th International Colloquium on Automata,
Languages and Programming (ICALP), pp.2--6,
Springer LNCS 1256, 1997.

Mannila. Local and Global Methods in Data Mining:
Basic Techniques and Open Problems In: Proc. 29th
ICALP, pp.57--68, Springer LNCS 2380, 2002.

Eiter, Gottlob. Hypergraph Transversal Computation
and Related Problems in Logic and AI, In: Proc. European
Conference on Logics in Artificial Intelligence (JELIA),
pp. 549-564, Springer LNCS 2224, 2002

Best known

[Fredman, Khachiyan, 94]

$N^{o(\log N)}$ time where $N = |\varphi| + |\psi|$

[Eiter, Gottlob, Makino, 02]

$o(\log^2 N)$ guessed bits

Summary

Knowlege	Explanations E w.r.t. $A = Lit$	
	query q	query $\overline{q}$
Horn CNF	P delay	no output P
Acyclic Horn CNF	P delay	incremental P
Characteristic set	MDual	MDual

Knowlege	Explanations E w.r.t. $A \subseteq Lit$	
	query q	query $\overline{q}$
Horn CNF	coNPc	coNPc
Acyclic Horn CNF	coNPc	coNPc
Characteristic set	MDual	MDual

Explanations E w.r.t. $A = Lit$

query / Knowlege	general	DNF	CNF	clause pos	clause Horn	clause general	term pos	term neg	term general
Horn CNF	coNPc	nOP	coNPc	Pd	coNPc	coNPc	Pd	nOP	nOP
$char(\Sigma)$	coNPc	nOP	coNPc	nOP	MD	nOP	MD	nOP	nOP

Explanations E w.r.t. $A \subseteq Lit$

query / Knowlege	general	DNF	CNF	clause pos	clause Horn	clause general	term pos	term neg	term general
Horn CNF	Π_2^P	Π_2^P	coNPc		coNPc			coNPc	
$char(\Sigma)$	Π_2^P	Π_2^P	coNPc	nOP	MD	nOP	MD	coNPc	coNPc

nOP: no Output P, Pd: P delay, MD: Monotone Dual

Open Problems

1. Abductive Inference

2. Monotone Dualization

3. Horn Transformation

4. Vertex Enumeration

Conclusion

Generating abductive explanations from Horn CNFs

Practical side

High order logic, non-Horn case.

Procedure Resolution

Input: A CNF $\varphi = \bigwedge_{i=1}^{m} c_i$ representing f.

Output: All prime implicates of f.

Step 1: $S := \{c_i \mid i = 1, 2, \cdots, m\}$.

Step 2: Repeat (s) simplification and (r) resolution.

(s) Remove c from S if $\exists c^* \in S$ s.t. $c^* \models c$.

(r) Add a resolvent c_3 of two clauses c_1, c_2 in S.

Step 3: Output all clauses in S.

Modification

(1) Input resolution: $c_1 \in \varphi$

(2) Add a prime implicate c' s.t. $c' \models c_3$

(3) Output c' in (r) immediately, if c' is new.

(4) $c_2 \ni q$

prime

Sketch: All explanations can be generated by the input resolution procedure if φ is definite Horn

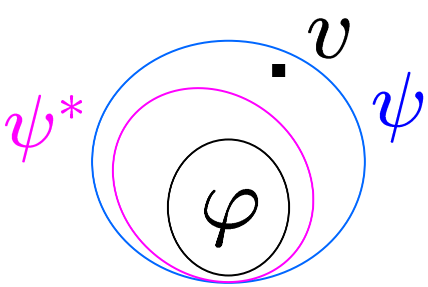
ψ^*: CNF consisting of all prime implicates $\ni q$

No negative clause $(\overline{x_1} \vee \overline{x_3} \vee \overline{x_2})$

ψ: CNF consisting of all prime implicates $\ni q$ generated so far

Note ψ^*: irredundant ($\forall c \in \varphi^* : \psi^* \not\equiv \psi^* \setminus \{c\}$)

if $\psi \neq \psi^*$ $\quad \exists v : \psi(v) = 1, \psi^*(v) = 0$



$\boxed{v\text{: maximal}}$

$\exists c \in \varphi, \exists c' \in \psi^* \setminus \psi : c(v) = c'(v) = 0$

$c = (p \vee \bigvee_{x \in N} \overline{x})$

$c' = (q \vee \bigvee_{x \in N'} \overline{x})$

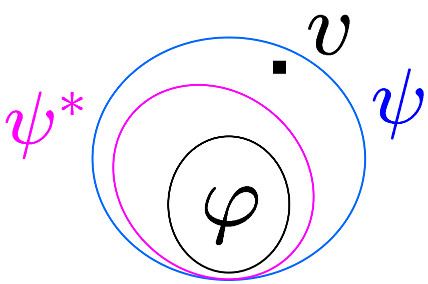
$$v = (\underset{q}{0}, \underset{p}{0}, \dots, 0, \overbrace{1, \dots,}^{N'} \underbrace{\dots, 1}_{N})$$

$$\boxed{p \neq q}$$

ψ^*: CNF consisting of all prime implicates $\ni q$

ψ : CNF consisting of all prime implicates $\ni q$ generated so far

if $\psi \neq \psi^*$ $\quad \exists v : \psi(v) = 1, \psi^*(v) = 0$



$\boxed{v\text{: maximal}}$

$\exists c \in \varphi, \exists c' \in \psi^* \setminus \psi : c(v) = c'(v) = 0$

$$v = (\underset{q}{0}, \underset{p}{0}, \ldots, 0, \overbrace{1, \ldots}^{N'}, \underbrace{\ldots, 1}_{N})$$

$$c = (p \vee \bigvee_{x \in N} \overline{x})$$

$$c' = (q \vee \bigvee_{x \in N'} \overline{x})$$

$$v + e_p = (0, 1, \ldots, 0, 1, \ldots, 1)$$

$$c'(v + e_p) = 0, \quad \psi^*(v + e_p) = 0, \quad \psi(v + e_p) = 0$$

$$\exists c'' \in \psi : c''(v + e_p) = 0$$

Since $c''(v) = 1$, $c'' = (q \vee \bigvee_{x \in N'' \cup \{p\}} \overline{x})$

$\boxed{\text{Resolution of } c \text{ and } c''}$

ψ

$\boldsymbol{q}$

Proof. ($mod(f)$ is closed under $\wedge$)

$$
\begin{array}{rl}
 & \quad\;\; I_1 \qquad\quad I_2 \qquad\quad I_3 \qquad\quad I_4 \\
v = & (1 \cdots 1 \mid 1 \cdots 1 \mid 0 \cdots 0 \mid 0 \cdots 0) \\
w = & (1 \cdots 1 \mid 0 \cdots 0 \mid 1 \cdots 1 \mid 0 \cdots 0) \\
v \wedge w = & (1 \cdots 1 \mid 0 \cdots 0 \mid 0 \cdots 0 \mid 0 \cdots 0)
\end{array}
$$

$\exists$ Horn clause $c = \bigwedge_{i \in N} \overline{x}_i \wedge x_j$:

$$c(v) = c(w) = 1,\ c(v \wedge w) = 0$$

From $c(v \wedge w) = 0,\ N \subseteq I_1,\ j \in I_2 \cup I_3 \cup I_4$

$c(v) = 0$ or $c(w) = 0$. a contradiction.