Knowledge Representation and Reasoning
Logics for Artificial Intelligence

Stuart C. Shapiro

Department of Computer Science and Engineering
and Center for Cognitive Science
University at Buffalo, The State University of New York
Buffalo, NY 14260-2000

shapiro@cse.buffalo.edu

copyright ©1995, 2004–2009 by Stuart C. Shapiro

Contents

Part I

Part II

Part III

14 Abduction

Abduction is the non-sound inference

from

$P \Rightarrow Q$

and Q

to

P

See *Brachman & Levesque*, Chapter 13.

Some Uses of Abduction

1. Explanation

   from $It's\ raining \Rightarrow The\ grass\ is\ wet$

   and $The\ grass\ is\ wet$ to $It's\ raining$

2. Diagnosis

   from $Infection \Rightarrow Fever$

   and $Fever$ to $Infection$

3. Plan Recognition

   from $Cooking\ pasta \Rightarrow Boil\ water$

   and $Boil\ water$ to $Cooking\ pasta$

4. Text Understanding

   from $\forall x(gotGoodService(x) \Rightarrow leftBigTip(x))$

   and *Betty left a big tip.* to *Betty got good service.*

Prime Implicates

Applies to KRR using resolution.

For some KB and some clause C, if
KB $\models C$
and for any $C\prime$ s.t. $C\prime$ is a proper subset of C
KB $\not\models C\prime$
C is a prime implicate of KB.

Example of Computing Prime Implicate

```
prover(4): (prove '( (=> (and p q r) g)
                     (=> (and (not p) q) g)
                     (=> (and (not q) r) g))
                  'g)

 1  (p (not q) g)                        Assumption
 2  (q (not r) g)                        Assumption
 3  ((not p) (not q) (not r) g)          Assumption
 4  ((not g))                            From Query

 5  (p (not q))                 R,4,1,{}
 6  (q (not r))                 R,4,2,{} Subsumed
 7  ((not p) (not q) (not r)) R,4,3,{} Subsumed
 8  ((not r) p)                 R,5,6,{} Subsumed
11  ((not q) (not r))           R,7,8,{} Subsumed
12  ((not r))                   R,11,6,{}
```

Example from *Brachman & Levesque*, p 271.

Example 2

```
prover(8): (prove '((forall x (=> (enterRestaurant x) (beSeated x)))
                    (forall x (=> (beSeated x) (beServed x)))
                    (forall x (=> (beServed x) (getFood x)))
                    (forall x (=> (getFood x) (eatFood x)))
                    (forall x (=> (eatFood x) (and (pay x) (leaveTip x))))
                    (forall x (=> (gotGoodService x) (leftBigTip x)))
                    (enterRestaurant Betty))
                  '(leftBigTip Betty))

 1  ((enterRestaurant Betty)) Assumption
 2  ((not (enterRestaurant ?1)) (beSeated ?1))  Assumption
 3  ((not (beSeated ?3)) (beServed ?3)) Assumption
 4  ((not (beServed ?5)) (getFood ?5))  Assumption
 5  ((not (getFood ?7)) (eatFood ?7)) Assumption
 6  ((not (eatFood ?9)) (pay ?9)) Assumption
 7  ((not (eatFood ?10)) (leaveTip ?10))  Assumption
 8  ((not (gotGoodService ?12)) (leftBigTip ?12)) Assumption
 9  ((not (leftBigTip Betty)) (Answer (leftBigTip Betty)))  From Query
10  ((not (gotGoodService Betty))
     (Answer (leftBigTip Betty)))  R,9,8,{Betty/?12}
nil
```

I.e., `(=> (gotGoodService Betty) (leftBigTip Betty))`

Interpretation

Possible interpretations of
`(=> (gotGoodService Betty) (leftBigTip Betty))`:

1. Abduction: Since `(leftBigTip Betty)`,
   infer `(gotGoodService Betty)`.
2. Diagnosis: Since `(not (leftBigTip Betty))`,
   infer `(not (gotGoodService Betty))`.
3. Hypothetical Answer: If `(gotGoodService Betty)`
   then `(leftBigTip Betty)`.
4. Why Not: Didn't infer `(leftBigTip Betty)`
   because didn't know `(gotGoodService Betty)`.