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, 2004–2010 by Stuart C. Shapiro

Contents

Part I

Part II

Part III

9 Belief Revision/Truth Maintenance

9.1 Motivation
Floors Above and Below Ground

```
: xor{OnFloor(1),OnFloor(2),OnFloor(3),OnFloor(4)}.
: {OnFloor(1), OnFloor(2)} => {Location(belowGround)}.
: {OnFloor(3), OnFloor(4)} => {Location(aboveGround)}.

: perform believe(OnFloor(1))

: list-asserted-wffs
  wff13!:  ~OnFloor(2)
  wff12!:  ~OnFloor(3)
  wff11!:  ~OnFloor(4)
  wff9!:  {OnFloor(4),OnFloor(3)} v=> {Location(aboveGround)}
  wff7!:  {OnFloor(2),OnFloor(1)} v=> {Location(belowGround)}
  wff6!:  Location(belowGround)
  wff5!:  xor{OnFloor(4),OnFloor(3),OnFloor(2),OnFloor(1)}
  wff1!:  OnFloor(1)
```

Motivation
Disbelieving an Hypothesis

```
: perform disbelieve(OnFloor(1))

: list-asserted-wffs
  wff9!:  {OnFloor(4),OnFloor(3)} v=> {Location(aboveGround)}
  wff7!:  {OnFloor(2),OnFloor(1)} v=> {Location(belowGround)}
  wff5!:  xor{OnFloor(4),OnFloor(3),OnFloor(2),OnFloor(1)}
```

Note the absence of `Location(belowGround)`

Moral

If retain derived beliefs (lemmas),

need a way to delete them
when their foundations are removed.

When Needed 1

If the KB contains beliefs about the (some) world,

and the world changes,

and the KB does not have a model of time.

I.e. the beliefs in the KB are of the form,
I believe this is true now.

What's needed

Links from hypotheses to propositions derived from them.

=> *vs.* `when(ever)do`: Assertions

```
: Floor({1,2,3,4}).
: xor{OnFloor(1),OnFloor(2),OnFloor(3),OnFloor(4)}.
: {OnFloor(1), OnFloor(2)} => {Location(belowGround)}.
: {OnFloor(3), OnFloor(4)} => {Location(aboveGround)}.
: perform withall(f, Floor(f),
                  adopt(wheneverdo(OnFloor(f),
                        believe(HaveBeenOnFloor(f)))),
                  noop()).
: perform believe(OnFloor(1))
```

=> *vs.* when(ever)do: The KB

```
: list-asserted-wffs
  wff37!:  ~OnFloor(2)
  wff36!:  ~OnFloor(3)
  wff35!:  ~OnFloor(4)
  wff31!:  wheneverdo(OnFloor(4),believe(HaveBeenOnFloor(4)))
  wff27!:  wheneverdo(OnFloor(3),believe(HaveBeenOnFloor(3)))
  wff23!:  wheneverdo(OnFloor(2),believe(HaveBeenOnFloor(2)))
  wff19!:  wheneverdo(OnFloor(1),believe(HaveBeenOnFloor(1)))
  wff17!:  HaveBeenOnFloor(1)
  wff16!:  Floor(1)
  wff15!:  Floor(2)
  wff14!:  Floor(3)
  wff13!:  Floor(4)
  wff10!:  {OnFloor(4),OnFloor(3)} v=> {Location(aboveGround)}
  wff8!:  {OnFloor(2),OnFloor(1)} v=> {Location(belowGround)}
  wff7!:  Location(belowGround)
  wff6!:  xor{OnFloor(4),OnFloor(3),OnFloor(2),OnFloor(1)}
  wff2!:  OnFloor(1)
  wff1!:  Floor({4,3,2,1})
```

=> *vs.* when(ever)do: Move Floors

```
: perform believe(OnFloor(4))

: list-asserted-wffs
  wff39!:  ~OnFloor(1)
  wff37!:  ~OnFloor(2)
  wff36!:  ~OnFloor(3)
  wff31!:  wheneverdo(OnFloor(4),believe(HaveBeenOnFloor(4)))
  wff29!:  HaveBeenOnFloor(4)
  wff27!:  wheneverdo(OnFloor(3),believe(HaveBeenOnFloor(3)))
  wff23!:  wheneverdo(OnFloor(2),believe(HaveBeenOnFloor(2)))
  wff19!:  wheneverdo(OnFloor(1),believe(HaveBeenOnFloor(1)))
  wff17!:  HaveBeenOnFloor(1)
  wff16!:  Floor(1)
  wff15!:  Floor(2)
  wff14!:  Floor(3)
  wff13!:  Floor(4)
  wff10!:  {OnFloor(4),OnFloor(3)} v=> {Location(aboveGround)}
  wff9!:  Location(aboveGround)
  wff8!:  {OnFloor(2),OnFloor(1)} v=> {Location(belowGround)}
  wff6!:  xor{OnFloor(4),OnFloor(3),OnFloor(2),OnFloor(1)}
  wff5!:  OnFloor(4)
  wff1!:  Floor({4,3,2,1})
```

`HaveBeenOnFloor(1)` remains; `OnFloor(1)` doesn't.

Moral

The consequents of
`=>`, `v=>`, `&=>`, `or`, `nand`, `xor`, `iff`, `andor`, `thresh`, and `nexists`
are derived and retain a connection to their underlying hypotheses.

Whatever is `believe`'d is a hypothesis.

Use `=>`, `v=>`, `&=>`, `or`, `nand`, `xor`, `iff`, `andor`, `thresh`, and `nexists`
for logical implications.

Use `whendo(p1,believe(p2))` or `wheneverdo(p1,believe(p2))`
for decisions.

Contingent Plans

```
: xor{Location(BellHall), Location(home)}.
: Location(BellHall) => ActPlan(getMail, go(MailRoom)).
: Location(home) => ActPlan(getMail, go(mailBox)).

: perform believe(Location(BellHall))
: ActPlan(getMail, ?how)?
  wff5!:  ActPlan(getMail,go(MailRoom))

: perform believe(Location(home))
: ActPlan(getMail, ?how)?
  wff8!:  ActPlan(getMail,go(mailBox))
```

Moral

Using this design for contingent plans,

along with retention of lemmas,

depends on belief revision.

Motivation
Sea Creatures

```
: all(x)(andor(0,1){Ako(x, mammal), Ako(x, fish)}).

: all(x)(LiveIn(x, water) => Ako(x, fish)).

: all(x)(BearYoung(x, live) => Ako(x, mammal)).

: LiveIn(whales, water).
: LiveIn(sharks, water).

: BearYoung(whales, live).
: BearYoung(dogs, live).
```

Motivation
Are Whales Fish or Mammals?

```
: Ako(whales, ?x)?

A contradiction was detected within context default-defaultct
The contradiction involves the newly derived proposition:
    wff8!:  Ako(whales,mammal)

and the previously existing proposition:
    wff9!:  ~Ako(whales,mammal)
```

SNeBR Options

```
You have the following options:
 1. [C]ontinue anyway, knowing that a contradiction is derivabl
 2. [R]e-start the exact same run in a different context which
    not inconsistent;
 3. [D]rop the run altogether.

  (please type c, r or d)
=><= r

In order to make the context consistent you must delete at leas
one hypothesis from each of the following sets of hypotheses:
  (wff6 wff4 wff3 wff2 wff1)
```

Possible Culprits

```
 In order to make the context consistent you must delete
  at least one hypothesis from the set listed below.
An inconsistent set of hypotheses:
 1 :   wff6!:  BearYoung(whales,live)
          (2 supported propositions: (wff8 wff6) )
 2 :   wff4!:  LiveIn(whales,water)
          (3 supported propositions: (wff10 wff9 wff4) )
 3 :   wff3!:  all(x)(BearYoung(x,live) => Ako(x,mammal))
          (2 supported propositions: (wff8 wff3) )
 4 :   wff2!:  all(x)(LiveIn(x,water) => Ako(x,fish))
          (3 supported propositions: (wff10 wff9 wff2) )
 5 :   wff1!:  all(x)(nand{Ako(x,fish),Ako(x,mammal)})
          (2 supported propositions: (wff9 wff1) )
```

Choosing the Culprit

```
  Enter the list number of a hypothesis to examine or
  [d] to discard some hypothesis from this list,
  [a] to see ALL the hypotheses in the full context,
  [r] to see what you have already removed,
  [q] to quit revising this set, or
  [i] for instructions

  (please type a number OR d, a, r, q or i)
=><= d

  Enter the list number of a hypothesis to discard,
  [c] to cancel this discard, or [q] to quit revising this set.
=><= 4
```

Remaining Possible Culprits

```
The consistent set of hypotheses:
 1 :   wff6!:  BearYoung(whales,live)
          (2 supported propositions: (wff8 wff6) )

 2 :   wff4!:  LiveIn(whales,water)
          (1 supported proposition: (wff4) )

 3 :   wff3!:  all(x)(BearYoung(x,live) => Ako(x,mammal))
          (2 supported propositions: (wff8 wff3) )

 4 :   wff1!:  all(x)(nand{Ako(x,fish),Ako(x,mammal)})
          (1 supported proposition: (wff1) )
  Enter the list number of a hypothesis to examine or
  [d] to discard some hypothesis from this list,
  [a] to see ALL the hypotheses in the full context,
  [r] to see what you have already removed,
  [q] to quit revising this set, or
  [i] for instructions
  (please type a number OR d, a, r, q or i)
=><= q
```

Other Hypotheses

```
  The following (not known to be inconsistent) set of
  hypotheses was also part of the context where the
  contradiction was derived:
    (wff7 wff5)
  Do you want to inspect or discard some of them?
=><= no
  Do you want to add a new hypothesis?  no

  wff11!:  ~Ako(whales,fish)
  wff8!:  Ako(whales,mammal)

 CPU time : 0.03
```

Resultant KB

```
: list-asserted-wffs
  wff12!:  ~(all(x)(LiveIn(x,water) => Ako(x,fish)))
  wff11!:  ~Ako(whales,fish)
  wff8!:  Ako(whales,mammal)
  wff7!:  BearYoung(dogs,live)
  wff6!:  BearYoung(whales,live)
  wff5!:  LiveIn(shakes,water)
  wff4!:  LiveIn(whales,water)
  wff3!:  all(x)(BearYoung(x,live) => Ako(x,mammal))
  wff1!:  all(x)(nand{Ako(x,fish),Ako(x,mammal)})
```

Moral
When Needed 2

If accepting information from multiple sources,

or just one possibly inconsistent source,

need a way to recognize contradictions,

and to find the culprit,

and to delete it,

and its implications.

What's Needed

Links between derived propositions

and hypotheses they were derived from.

9.2 Relevance Logic (R)

Motivation

Paradoxes of Implication 1

Anything Implies a Truth

1	A	Hyp
2	B	Hyp
3	A	Reit, 1
4	$B \Rightarrow A$	$\Rightarrow$ I, 2–3
5	$A \Rightarrow (B \Rightarrow A)$	$\Rightarrow$ I, 1–4

But it seems that B had nothing to do with deriving A.

Motivation of R
Paradoxes of Implication 2
A Contradiction Implies Anything

1	$A \wedge \neg A$	Hyp
2	$\neg B$	Hyp
3	$A \wedge \neg A$	Reit, 1
4	A	$\wedge$E, 3
5	$\neg A$	$\wedge$E, 3
6	B	$\neg$I, 2–5
7	$(A \wedge \neg A) \Rightarrow B$	$\Rightarrow$I, 1–6

But it seems that $\neg B$ had nothing to do with deriving the contradiction.

What's Needed

A way to determine when a hypothesis is really used to derive another wff.

When a hypothesis is **relevant** to a conclusion.

9.3 R
Relevance Logic
The Logic of Relevant Implication

Syntax: The same as Standard FOL.

Intensional Semantics: The same as Standard FOL.

Extensional Semantics: The same as Standard FOL for terms.
For wffs: a four-valued logic, using True, False, Neither, and Both.

KB Interpretations of R's 4 Truth Values

True	true
False	false
Neither	unknown
Both	contradictory, "I've been told both."
	or a "true contradiction"
	such as Russell's set both is and isn't a member of itself.

9.4 R Proof Theory

Structural Rules of Inference

$$\begin{array}{r|ll} i. & A, \{n\} & Hyp \\ \hline & & \\ \end{array}$$

$$\begin{array}{r|ll} i. & A, \alpha & \\ & \vdots & \\ j. & A, \alpha & Rep, i \end{array}$$

$$\begin{array}{r|l} i. & A, \alpha \\ & \vdots \\ & \quad . \\ & \qquad . \\ & \qquad\quad . \\ & \quad \left| \; \vdots \right. \\ j. & \quad \left| \; A, \alpha \quad Reit, i \right. \end{array}$$

where n is a new integer.

R Rules for $\Rightarrow$

$i.$	$\quad A, \{n\}$	Hyp
	$\quad \vdots$	
$j.$	$\quad B, \alpha, \ s.t.\ n \in \alpha$	
k	$(A \Rightarrow B), \alpha - \{n\}$	$\Rightarrow I, i\text{–}j$

$i.$	A, α	
	$\vdots$	
$j.$	$(A \Rightarrow B), \beta$	
$k.$	$B, \alpha \cup \beta$	$\Rightarrow E, i, j$

How the Paradoxes of Implication are Blocked 1

1.	$A, \{1\}$	*Hyp*
2.	$B, \{2\}$	*Hyp*
3.	$A, \{1\}$	*Reit*, 1

Can't then apply $\Rightarrow I$

R Rules for $\wedge$

$i_1.$	A_1, α
	$\vdots$
$i_n.$	A_n, α
$j.$	$A_1 \wedge \cdots \wedge A_n, \alpha \quad \wedge I, i_1, \ldots, i_n$

$i.$	$A_1 \wedge \cdots \wedge A_n, \alpha$
	$\vdots$
$j.$	$A_k, \alpha \qquad \wedge E, i$

Why $\wedge I$ Requires the Same OS If Not

1	$A, \{1\}$	Hyp, 2–5
2	$B, \{2\}$	Hyp, 3–5
3	$A, \{1\}$	Reit, 1
4	$(A \wedge B), \{1, 2\}$	$\wedge$I?
5	$A, \{1, 2\}$	$\wedge$E, 4
6	$(B \Rightarrow A), \{1\}$	$\Rightarrow$I, 2–5
7	$(A \Rightarrow (B \Rightarrow A)), \{\}$	$\Rightarrow$I, 1–6

Reconstruct paradox of implication.

Note: Empty os means a theorem.

Extended Rule for $\wedge I$

$$
\begin{array}{r|ll}
i_1. & A_1, \alpha & \\
 & \vdots & \\
i_n. & A_n, \eta & \\
j. & A_1 \wedge \cdots \wedge A_n, (\alpha \cup \cdots \cup \eta)^* & \wedge I, i_1, \ldots, i_n
\end{array}
$$

Can't apply $\wedge E$ to an extended wff.

R Rules for ¬

$i.$	$A, \{n\}$	Hyp
	$\vdots$	
$j.$	$B, \alpha \ s.t. \ n \in \alpha$	
$j+1.$	$\neg B, \alpha$	
$j+2.$	$\neg A, \alpha - \{n\}$	$\neg I, i\text{–}(j+1)$

$i.$	$\neg A, \{n\}$	Hyp
	$\vdots$	
$j.$	$B, \alpha \ s.t. \ n \in \alpha$	
$j+1.$	$\neg B, \alpha$	
$j+2.$	$A, \alpha - \{n\}$	$\neg I, i\text{–}(j+1)$

$i.$	$\neg\neg A, \alpha$	
$j.$	A, α	$\neg E, i$

Extended R Rule for $\neg I$

$i.$	$A, \{n\}$	Hyp
	$\vdots$	
$j.$	B, α	
$j+1.$	$\neg B, \beta$	
$j+2.$	$\neg A, ((\alpha \cup \beta) - \{n\})^*$ $s.t.$ $n \in (\alpha \cup \beta)$	$\neg I, i$–$(j+1)$

$i.$	$\neg A, \{n\}$	Hyp
	$\vdots$	
$j.$	B, α	
$j+1.$	$\neg B, \beta$	
$j+2.$	$A, ((\alpha \cup \beta) - \{n\})^*$ $s.t.$ $n \in (\alpha \cup \beta)$	$\neg I, i$–$(j+1)$

How the Paradoxes of Implication are Blocked 2

1.	$(A \wedge \neg A), \{1\}$	*Hyp*
2.	$\neg B, \{2\}$	*Hyp*
3.	$(A \wedge \neg A), \{1\}$	*Reit*, *1*
4.	$A, \{1\}$	$\wedge E$, *3*
5.	$\neg A, \{1\}$	$\wedge E$, *3*

Can't then apply $\neg I$

R is a **paraconsistent** logic:

a contradiction does not imply anything whatsoever.

R Rule for $\vee I$

$$
\begin{array}{r|ll}
i. & A_i, \alpha & \\
j. & A_1 \vee \cdots \vee A_i \vee \cdots \vee A_n, \alpha & \vee I, i
\end{array}
$$

R Rule for $\vee E$

$i_1.$	$A_1 \vee \cdots \vee A_n, \alpha$	
	$\vdots$	
$i_2.$	$A_1 \Rightarrow B, \beta$	
	$\vdots$	
$i_3.$	$A_n \Rightarrow B, \beta$	
$j.$	$B, \alpha \cup \beta$	$\vee E, i_1, i_2, i_3$

Irrelevance of Disjunctive Syllogism

1	$((A \vee B) \wedge \neg A), \{1\}$	Hyp	
2	$\neg A, \{1\}$	$\wedge$E,1	
3	$(A \vee B), \{1\}$	$\wedge$E,1	
4	$A, \{2\}$	Hyp	
5	$\neg B, \{3\}$	Hyp	
6	$A, \{2\}$	Reit, 4	
7	$\neg A, \{1\}$	Reit, 2	
8	B	$\neg$I, 5–7	**Not valid in R**
9	$A \Rightarrow B$	$\Rightarrow$I, 4–8	
10	$B, \{4\}$	Hyp	
11	$B, \{4\}$	Rep, 10	
12	$B \Rightarrow B, \{\}$	$\Rightarrow$I, 10–11	
13	B	$\vee$E, 3,9,12	

So $\vee$ is just truth-functional.

R Rules for Intensional OR ($\oplus$)

$$
\begin{array}{r|ll}
i. & (\neg A \Rightarrow B), \alpha & \\
 & \vdots & \\
j. & (\neg B \Rightarrow A), \alpha & \\
j+1. & (A \oplus B), \alpha & \oplus I, i, j
\end{array}
$$

$$
\begin{array}{r|ll}
i. & (A \oplus B), \alpha & \\
 & \vdots & \\
j. & \neg A, \beta & \\
j+1. & B, \alpha \cup \beta & \oplus E
\end{array}
\qquad
\begin{array}{r|ll}
i. & (A \oplus B), \alpha & \\
 & \vdots & \\
j. & \neg B, \beta & \\
j+1. & A, \alpha \cup \beta & \oplus E
\end{array}
$$

R Rules for $\Leftrightarrow$

$$
\begin{array}{r|ll}
i. & (A \Rightarrow B), \alpha & \\
 & \vdots & \\
j. & (B \Rightarrow A), \alpha & \\
j+1. & (A \Leftrightarrow B), \alpha & \Leftrightarrow I, i, j
\end{array}
$$

$$
\begin{array}{r|ll}
i. & A, \alpha & \\
 & \vdots & \\
j. & (A \Leftrightarrow B), \beta & \\
j+1. & B, \alpha \cup \beta & \Leftrightarrow E, i, j
\end{array}
\qquad
\begin{array}{r|ll}
i. & B, \alpha & \\
 & \vdots & \\
j. & (A \Leftrightarrow B), \beta & \\
j+1. & A, \alpha \cup \beta & \Leftrightarrow E, i, j
\end{array}
$$

R Rules for $\forall$

$i.$	$A(a), \{n\}$	Hyp
	$\vdots$	
$j.$	$B(a), \alpha \; s.t. \; n \in \alpha$	
$j+1.$	$\forall x(A(x) \Rightarrow B(x)), \alpha - \{n\}$	$\forall I, i\text{–}j$

$i.$	$A(t), \alpha$	
	$\vdots$	
$j.$	$\forall x(A(x) \Rightarrow B(x)), \beta$	
$j+1.$	$B(t), \alpha \cup \beta$	$\forall E, i, j$

Where a is an arbitrary individual not otherwise used in the proof, and t is free for x in $B(x)$.

Note $\forall$ only governs $\Rightarrow$.

R Rules for $\exists$

$$
\begin{array}{r|ll}
i & A(t), \alpha & \\
i+1 & \exists x A(x), \alpha & \exists I, i
\end{array}
$$

$$
\begin{array}{r|lll}
i & & \exists x A(x), \alpha & \\
 & & \vdots & \\
j & \quad| & A\{a/x\}, \beta & \text{Indef I}, i \\
\cline{3-4}
 & \quad| & \vdots & \\
k & \quad| & B, \gamma \ s.t. \ \beta \subset \gamma & \\
k+1 & & B, \gamma - \beta & \exists E, j\text{–}k
\end{array}
$$

Where $A(x)$ is the result of replacing some or all occurrences of t in $A(t)$ by x,
t is free for x in $A(x)$;
a is an indefinite individual not otherwise used in the proof,
$A(a/x)$ is the result of replacing all occurrences of x in $A(x)$ by a,
and there is no occurrence of a in B.

Why the Subproof Contours?

1. To keep track of assumptions for each derived wff.
   But this is accomplished by os.

2. To differentiate hypotheses from derived wffs.
   Introduce support: $\langle\{hyp \mid der \mid ext\}, os\rangle$
   with origin tag and origin set.

SNePS KB

The SNePS KB consists of a collection of supported wffs.

A wff may have more than one support if it was derived in multiple ways.

Every implemented rule of inference specifies how the derived wff is derived from its parent(s) and how its support is derived from the support(s) of its parent(s).

Contexts and Belief Spaces

A context is a set of hypotheses.

A belief space defined by a context c is the set containing every wff whose os is a subset of c.

SNePSLOG Example

```
: expert
...
: xor{OnFloor(1),OnFloor(2),OnFloor(3),OnFloor(4)}.
  wff5!:  xor{OnFloor(4),OnFloor(3),OnFloor(2),OnFloor(1)}
          {<hyp,{wff5}>}

: {OnFloor(1), OnFloor(2)} => {Location(belowGround)}.
  wff7!:  {OnFloor(2),OnFloor(1)} v=> {Location(belowGround)}
          {<hyp,{wff7}>}

: {OnFloor(3), OnFloor(4)} => {Location(aboveGround)}.
  wff9!:  {OnFloor(4),OnFloor(3)} v=> {Location(aboveGround)}
          {<hyp,{wff9}>}
```

```
: perform believe(OnFloor(1))

: describe-context
((assertions (wff9 wff7 wff5 wff1))
 (named (default-defaultct)) (kinconsistent nil))
```

```
: list-asserted-wffs
  wff13!:  ~OnFloor(2)  {<der,{wff1,wff5}>}
  wff12!:  ~OnFloor(3)  {<der,{wff1,wff5}>}
  wff11!:  ~OnFloor(4)  {<der,{wff1,wff5}>}
  wff9!:  {OnFloor(4),OnFloor(3)} v=> {Location(aboveGround)}
          {<hyp,{wff9}>}
  wff7!:  {OnFloor(2),OnFloor(1)} v=> {Location(belowGround)}
          {<hyp,{wff7}>}
  wff6!:  Location(belowGround)  {<der,{wff1,wff7}>}
  wff5!:  xor{OnFloor(4),OnFloor(3),OnFloor(2),OnFloor(1)}
          {<hyp,{wff5}>}
  wff1!:  OnFloor(1)  {<hyp,{wff1}>}
```

```
: perform disbelieve(OnFloor(1))

: describe-context
((assertions (wff9 wff7 wff5)) (named (default-defaultct))
 (kinconsistent nil))

: list-asserted-wffs
  wff9!:  {OnFloor(4),OnFloor(3)} v=> {Location(aboveGround)}
          {<hyp,{wff9}>}
  wff7!:  {OnFloor(2),OnFloor(1)} v=> {Location(belowGround)}
          {<hyp,{wff7}>}
  wff5!:  xor{OnFloor(4),OnFloor(3),OnFloor(2),OnFloor(1)}
          {<hyp,{wff5}>}
```

SNePSLOG Example of $\neg I$

```
wff5!:  BearYoung(whales,live)  {<hyp,{wff5}>}
wff4!:  LiveIn(whales,water)  {<hyp,{wff4}>}
wff3!:  all(x)(BearYoung(x,live) => Ako(x,mammal))
        {<hyp,{wff3}>}
wff2!:  all(x)(LiveIn(x,water) => Ako(x,fish))
        {<hyp,{wff2}>}
wff1!:  all(x)(nand{Ako(x,fish),Ako(x,mammal)})
        {<hyp,{wff1}>}
```

```
: Ako(whales, ?x)?
  A contradiction was detected within context default-defaultct.
  The contradiction involves the newly derived proposition:
      wff8!:  Ako(whales,mammal)  {<der,{wff3,wff5}>}

  and the previously existing proposition:
      wff9!:  ~Ako(whales,mammal)  {<der,{wff1,wff2,wff4}>}

...
  In order to make the context consistent you must delete at least
  one hypothesis from each of the following sets of hypotheses:
    (wff5 wff4 wff3 wff2 wff1)
```

The Culprit Set

```
1 :   wff5!:  BearYoung(whales,live)  {<hyp,{wff5}>}
         (2 supported propositions: (wff8 wff5) )

2 :   wff4!:  LiveIn(whales,water)  {<hyp,{wff4}>}
         (3 supported propositions: (wff9 wff7 wff4) )

3 :   wff3!:  all(x)(BearYoung(x,live) => Ako(x,mammal)) {<hyp,{wff3
         (2 supported propositions: (wff8 wff3) )

4 :   wff2!:  all(x)(LiveIn(x,water) => Ako(x,fish)) {<hyp,{wff2}>}
         (3 supported propositions: (wff9 wff7 wff2) )

5 :   wff1!:  all(x)(nand{Ako(x,fish),Ako(x,mammal)})
              {<hyp,{wff1}>}
         (2 supported propositions: (wff9 wff1) )
```

KB after deleting wff2

```
wff10!:  ~(all(x)(LiveIn(x,water) => Ako(x,fish)))
         {<ext,{wff1,wff3,wff4,wff5}>}
wff8!:  Ako(whales,mammal)  {<der,{wff3,wff5}>}
wff7!:  ~Ako(whales,fish)  {<der,{wff1,wff3,wff5}>}
wff5!:  BearYoung(whales,live)  {<hyp,{wff5}>}
wff4!:  LiveIn(whales,water)  {<hyp,{wff4}>}
wff3!:  all(x)(BearYoung(x,live) => Ako(x,mammal))
        {<hyp,{wff3}>}
wff1!:  all(x)(nand{Ako(x,fish),Ako(x,mammal)})
        {<hyp,{wff1}>}
```