Paraconsistent Attitudes

— A Logical Model of Inconsistent Information

Paraconsistent logic is a new branch of symbolic logic which concerns the logical modelling of inconsistent information.

Classical logic predicts that everything (thus nothing useful at all) follows from inconsistency

In practice, 100% consistency is almost never achieved.

Examples of inconsistent information from which inferences are drawn include:

- The data presented to a jury in trial.
- The information fed into a computer.
- A person's set of beliefs.

The reasons for inconsistent information can be mistakes, multiple sources, etc.

Attitudes involve the attribution to persons or even computers of "mental things" like beliefs, fears and wishes.

Consider the situation illustrated with the drawing:

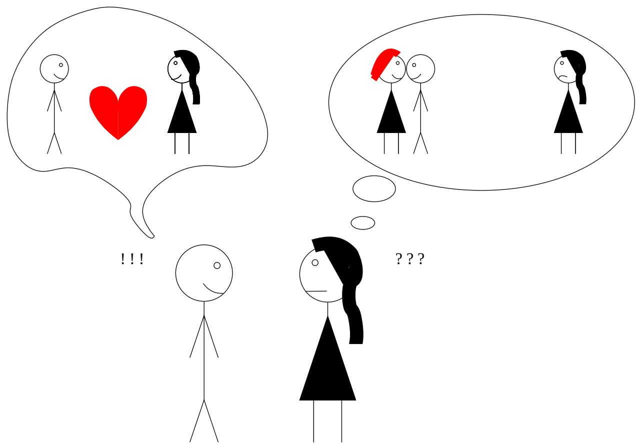
Bob asserts that Alice is his only love.

Alice believes that he loves Carol since he kissed her.

We note an inconsistency with the tacit assumption:

Alice believes Bob. (*)

Of course (*) can be revised, but such a simple solution is not always possible for practical or theoretical reasons.

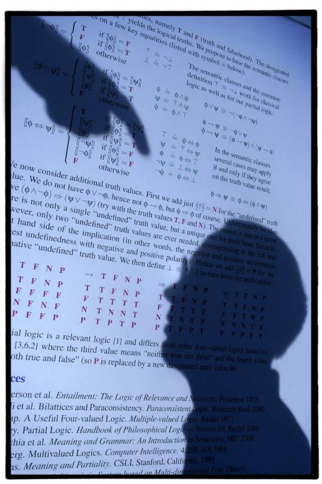


See also

Planned Workshop: *Paraconsistent Computational Logic*
Federated Logic Conference (Copenhagen 2002)
`http://floc02.diku.dk/PCL/`

Jørgen Villadsen: *Combinators for Paraconsistent Attitudes*
Lecture Notes in Computer Science 2099:261–278 (2001)
`http://www.imm.dtu.dk/~jv/poster/`

We base the semantic clauses for the logical operators on a few key equalities listed with symbol $\doteq$ below. The two truth values **T** and **F** stand for truth and falsehood, and we obtain a paraconsistent logic by allowing additional truth values.



$$[\![\neg\phi]\!] = \begin{cases} \mathbf{T} & \text{if } [\![\phi]\!] = \mathbf{F} \\ \mathbf{F} & \text{if } [\![\phi]\!] = \mathbf{T} \\ [\![\phi]\!] & \text{otherwise} \end{cases} \qquad \begin{array}{l} \top \doteq \neg\bot \\ \bot \doteq \neg\top \end{array}$$

$$[\![\phi \wedge \psi]\!] = \begin{cases} [\![\phi]\!] & \text{if } [\![\phi]\!] = [\![\psi]\!] \\ [\![\psi]\!] & \text{if } [\![\phi]\!] = \mathbf{T} \\ [\![\phi]\!] & \text{if } [\![\psi]\!] = \mathbf{T} \\ \mathbf{F} & \text{otherwise} \end{cases} \qquad \begin{array}{l} \phi \doteq \phi \wedge \phi \\ \psi \doteq \top \wedge \psi \\ \phi \doteq \phi \wedge \top \end{array} \qquad \phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$$

$$[\![\phi \Leftrightarrow \psi]\!] = \begin{cases} \mathbf{T} & \text{if } [\![\phi]\!] = [\![\psi]\!] \\ [\![\psi]\!] & \text{if } [\![\phi]\!] = \mathbf{T} \\ [\![\phi]\!] & \text{if } [\![\psi]\!] = \mathbf{T} \\ [\![\neg\psi]\!] & \text{if } [\![\phi]\!] = \mathbf{F} \\ [\![\neg\phi]\!] & \text{if } [\![\psi]\!] = \mathbf{F} \\ \mathbf{F} & \text{otherwise} \end{cases} \qquad \begin{array}{l} \top \doteq \phi \Leftrightarrow \phi \\ \psi \doteq \top \Leftrightarrow \psi \\ \phi \doteq \phi \Leftrightarrow \top \\ \neg\psi \doteq \bot \Leftrightarrow \psi \\ \neg\phi \doteq \phi \Leftrightarrow \bot \end{array} \qquad \phi \Rightarrow \psi \equiv \phi \Leftrightarrow (\phi \wedge \psi)$$

Informatics and Mathematical Modelling