Paraconsistent Computational Logic

Andreas Schmidt Jensen and Jørgen Villadsen

Algorithms and Logic Section, DTU Informatics, Denmark

Abstract. In classical logic everything follows from inconsistency and this makes classical logic problematic in areas of computer science where contradictions seem unavoidable. We describe a many-valued paraconsistent logic, discuss the truth tables and include a small case study.

1 Introduction — Motivation and Definitions

Often consistency cannot be assumed and a paraconsistent logic seems needed, in particular for applications of logic in computer science and artificial intelligence [1, 2]. We consider the propositional fragment of a higher-order paraconsistent logic [4, 5]. We have the two classical determinate truth values $\Delta = \{\bullet, \circ\}$ for truth and falsity and a countably infinite set of indeterminate truth values $\nabla = \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \ldots\}$. The only designated truth value $\bullet$ yields the logical truths. The indeterminate truth values are not at all ordered with respect to truth content. The logic is a generalization of Łukasiewicz's three-valued logic (originally proposed 1920–30), with the intermediate value duplicated many times and ordered such that none of the copies of this value imply other ones, but it differs from Łukasiewicz's many-valued logics as well as from logics based on bilattices [3]. The motivation for the logical operators is based on key equalities shown to the right of the semantic clauses (we also have $\varphi \Leftrightarrow \neg\neg\varphi$):

$$[\![\neg\varphi]\!] = \begin{cases} \bullet & \text{if } [\![\varphi]\!] = \circ \\ \circ & \text{if } [\![\varphi]\!] = \bullet \\ [\![\varphi]\!] & \text{otherwise} \end{cases} \qquad \begin{array}{l} \top \Leftrightarrow \neg\bot \\ \bot \Leftrightarrow \neg\top \end{array}$$

$$[\![\varphi \wedge \psi]\!] = \begin{cases} [\![\varphi]\!] & \text{if } [\![\varphi]\!] = [\![\psi]\!] \\ [\![\psi]\!] & \text{if } [\![\varphi]\!] = \bullet \\ [\![\varphi]\!] & \text{if } [\![\psi]\!] = \bullet \\ \circ & \text{otherwise} \end{cases} \qquad \begin{array}{l} \varphi \Leftrightarrow \varphi \wedge \varphi \\ \psi \Leftrightarrow \top \wedge \psi \\ \varphi \Leftrightarrow \varphi \wedge \top \end{array}$$

The basic semantic clause and the clause $[\![\top]\!] = \bullet$ are omitted. In the semantic clauses several cases may apply if and only if they agree on the result. Note that the semantic clauses work for classical logic too.

Abbreviations:

$$\bot \equiv \neg\top \qquad \varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$$

We continue with biimplication (and we then simply obtain implication and modality as abbreviations). The semantic clauses extend the clauses for equality $=$ which always give a determinate truth value:

$$[\![\varphi \Leftrightarrow \psi]\!] = \begin{cases} \bullet \text{ if } [\![\varphi]\!] = [\![\psi]\!] \\ \circ \text{ otherwise} \end{cases}$$

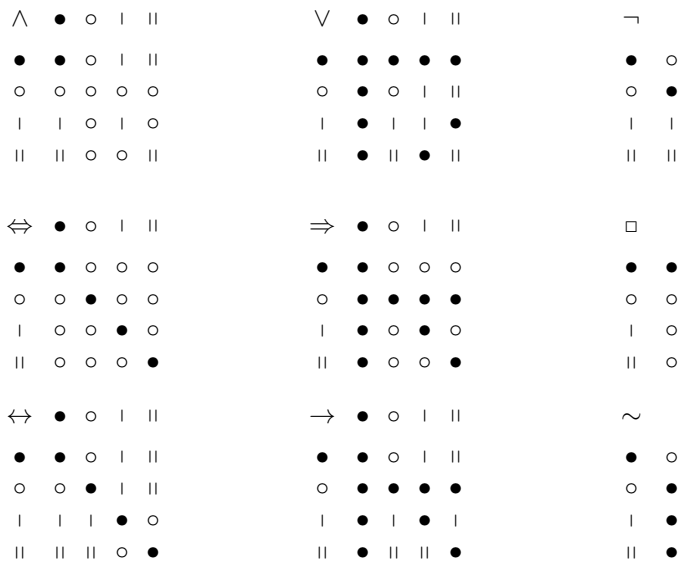
$$[\![\varphi \leftrightarrow \psi]\!] = \begin{cases} \bullet & \text{if } [\![\varphi]\!] = [\![\psi]\!] \\ [\![\psi]\!] & \text{if } [\![\varphi]\!] = \bullet \\ [\![\varphi]\!] & \text{if } [\![\psi]\!] = \bullet \\ [\![\neg\psi]\!] & \text{if } [\![\varphi]\!] = \circ \\ [\![\neg\varphi]\!] & \text{if } [\![\psi]\!] = \circ \\ \circ & \text{otherwise} \end{cases} \qquad \begin{array}{l} \top \Leftrightarrow \varphi \leftrightarrow \varphi \\ \psi \Leftrightarrow \top \leftrightarrow \psi \\ \varphi \Leftrightarrow \varphi \leftrightarrow \top \\ \neg\psi \Leftrightarrow \bot \leftrightarrow \psi \\ \neg\varphi \Leftrightarrow \varphi \leftrightarrow \bot \end{array}$$

Abbreviations:

$$\varphi \Rightarrow \psi \equiv \varphi \Leftrightarrow \varphi \wedge \psi \qquad \varphi \rightarrow \psi \equiv \varphi \leftrightarrow \varphi \wedge \psi \qquad \Box\varphi \equiv \varphi = \top \qquad \sim\varphi \equiv \neg\Box\varphi$$

2 Discussion of Truth Tables

Although we have a countably infinite set of truth value we can investigate the logic by truth tables since the indeterminate truth values are not ordered with respect to truth content.



The required number of indeterminacies corresponds to the number of propositions in a given formula. This also means that the logic is weakened when additional indeterminate truth values are added.

Given an atomic formula, it is clear that $\nabla = \{\text{I}\}$ suffices. To see this we use the fact that there exist no ordering between indeterminate truth values. If $[\![P]\!] = \text{II}$ and we replace the truth value with I then the truth value is still indeterminate. Now consider the tautology $P \to Q \leftrightarrow \neg Q \to \neg P$ (contraposition). Using a single indeterminacy yields no difference; the formula still holds. When using two indeterminacies we can give a counter-example:

$$P \to Q \leftrightarrow \neg\ Q \to \neg\ P$$
$$\text{I}\quad \text{I}\quad \text{II}\quad \circ\quad \text{II}\quad \text{II}\quad \text{II}\quad \text{I}\quad \text{I}$$

We could require $\nabla = \{\text{I}, \text{II}, \text{III}\}$, but the third indeterminacy is not needed since we already have one unique indeterminacy for each proposition.

As an example for three propositions, we can consider the formula $\neg(Q \to P) \to (\neg(\neg R \vee (R \to Q)) \to (P \to Q))$. It is a tautology when $\nabla = \emptyset$ (classical propositional logic), $\nabla = \{\text{I}\}$ and $\nabla = \{\text{I}, \text{II}\}$. When $\nabla = \{\text{I}, \text{II}, \text{III}\}$ it is no longer a tautology:

$$\neg\ (\ Q \to P\ ) \to (\ \neg\ (\ \neg\ R \vee (\ R \to Q\ )\ ) \to (\ P \to Q\ )\ )$$
$$\text{I}\quad \text{I}\quad \text{I}\quad \text{II}\quad \text{I}\quad \text{III}\quad \text{III}\ \text{III}\ \text{III}\quad \text{III}\ \text{III}\quad \text{I}\quad \text{III}\quad \text{II}\ \text{II}\quad \text{I}$$

3 Case Study

Paraconsistent logics are useful in areas where inconsistency is an acceptable feature of the systems involved. One such area is multi-agent systems since the belief base of an agent very well could contain contradictory beliefs and thus be inconsistent. Consider an agent with a set of beliefs (item 0) and rules:

0. $P \wedge Q \wedge \neg R$
1. $P \wedge Q \to R$
2. $R \to S$
3. $\neg S \to \neg R$

We can deduce R using classical logic. This leaves the agent with contradictory beliefs, namely R and $\neg R$. This entails everything, so the agent might start behaving in an undesirable way. It could now believe that $\neg P$, or $\neg Q$ – or even φ for any formula φ.

In our paraconsistent logic this is not the case. The following is not a tautology (the truth value of the main connective is ı):

$$(P \wedge Q \wedge \neg R) \wedge (P \wedge Q \to R) \to R$$
$$\bullet \ \imath \ \imath \ \imath \ \bullet \ \circ \quad \imath \quad \bullet \ \imath \ \imath \ \imath \ \circ \quad \imath \ \circ$$

Note that while rule 2 and 3 are classically equivalent (contraposition), it is not the case in our paraconsistent logic. This means that even if R follows, this does not necessarily mean that S follows as well.

We do however need some kind of modus ponens in order for the agent to be able to reason. We therefore require that rules are not allowed to be inconsistent and use the necessity operator to ensure either truth or falsity. This makes sense; after all, the agent requires absolute knowledge about whether its rules are applicable or not.

$$(P \wedge Q \wedge \neg R) \wedge \Box (P \wedge Q \to R) \Rightarrow R$$
$$\bullet \ \imath \ \imath \ \imath \ \bullet \ \circ \quad \circ \ \circ \quad \bullet \ \imath \ \imath \ \imath \ \circ \quad \bullet \ \circ$$

Note that applying the necessity operator on rules does not make the agent classical. $\neg P$ and $\neg Q$ still does not follow. We let $\rhd_{\text{XYZ}} P$ mean that P follows from the agents rules X, Y and Z.

$$(P \wedge Q \wedge \neg R) \wedge \Box(P \wedge Q \to R) \Rightarrow R \ \equiv \ \rhd_{01} \ R$$

The agent can then conclude the following from its beliefs and rules:

$$\not\rhd_{012} \neg P \quad \not\rhd_{012} \neg Q \quad \not\rhd_{012} \neg S$$
$$\rhd_{012} R \quad \rhd_{012} \neg R \quad \rhd_{012} S$$

We observe that the same follows when using rules 1 and 3, since the implication becomes classical when necessity is applied; $\Box(R \to S) \ \equiv \ \Box(\neg S \to \neg R)$. Note that using $\Rightarrow$ instead of $\Box$ and $\to$ would yield the same result.

This result differs from classical logic, where all the above propositions would follow from the rules and beliefs. The paraconsistent logic allows the agent to reason using inconsistent beliefs without entailing everything.

4 Conclusions

We have presented a paraconsistent logic defined using semantic clauses and motivated by key equalities. Although infinite-valued only a finite number of truth values must be considered for a given formula.

A small case study has been described and we have recently also investigated applications in logical semantics of natural language using a higher order logic extension with only propositional types [6].

References

1. D. Batens, C. Mortensen, G. Priest and J. Van-Bengedem (editors). *Frontiers in Paraconsistent Logic.* Research Studies Press, 2000.
2. H. Decker, J. Villadsen and T. Waragai (editors). *International Workshop on Paraconsistent Computational Logic.* Volume 95 of Roskilde University, Computer Science, Technical Reports, 2002.
3. S. Gottwald. *A Treatise on Many-Valued Logics.* Research Studies Press, 2001.
4. J. Villadsen. *A Paraconsistent Higher Order Logic.* Pages 38–51 in B. Buchberger and J. A. Campbell (editors), Springer Lecture Notes in Computer Science 3249, 2004.
5. J. Villadsen. *Supra-Logic: Using Transfinite Type Theory with Type Variables for Paraconsistency.* Journal of Applied Non-Classical Logics, 15(1):45–58, 2005.
6. J. Villadsen. *Infinite-Valued Propositional Type Theory for Semantics.* Pages 277–297 in J.-Y. Béziau and A. Costa-Leite (editors), Dimensions of Logical Concepts, Unicamp Coleç. CLE 54, 2009.