Investigations of Inferential Inconsistency

Michael Giancola

April 26, 2019

Contents

1 The Problem **3**

2 History of (In-)Consistency and ... **3**
- 2.1 Aristotle . . . 3
- 2.2 Logic Programming . . . 4
- 2.3 Computational Logic . . . 4
  - 2.3.1 Direct Logic . . . 4
  - 2.3.2 Logic Theorist . . . 5
- 2.4 Automated Reasoning . . . 5
- 2.5 AI . . . 5

3 Inferential Inconsistency Relevant to Gödel-Rosser **5**
- 3.1 Feferman 1984 . . . 5
- 3.2 Restall 2006 . . . 6

4 Relationship between (In-)Consistency and ... **7**
- 4.1 Defeasible Reasoning . . . 7
  - 4.1.1 Plan Construction . . . 7
- 4.2 Non-Monotonic Logic . . . 8
- 4.3 Truth-Maintenance Systems (TMSs) . . . 9
- 4.4 Belief Revision . . . 10
- 4.5 (Computational) Paraconsistent Logic . . . 11

5 Conclusions **12**

6 Appendix: Some Reflections **13**
- 6.1 Multiple Defeasible Arguments . . . 13
- 6.2 Contradictions in Knowledge Bases . . . 13

1 The Problem

Humans H need intelligent machines M to do important X for them, but M's doing X is imperiled by the presence of inferential inconsistency in the operation of M in pursuit of X, where the relevant operation of M must make crucial use of automated reasoning, and other activity based on automated reasoning (e.g. logic-based planning).

2 History of (In-)Consistency and ...

2.1 Aristotle

From what I've found so far, Aristotle didn't discuss the concepts of logical consistency or contradiction by name, but discusses them several times, more or less restating the Law of Non-Contradiction with different words, in slightly different contexts. He says in Prior Analytics [1], "it is impossible that the same thing should be necessitated by the being and by the not-being of the same thing." The following three quotations are from Metaphysics:

- "It is impossible that the same thing belong and not belong to the same thing at the same time and in the same respect."
- "No one can believe that the same thing can (at the same time) be and not be."
- "The most certain of all basic principles is that contradictory propositions are not true simultaneously."

He later[1] discussed the use of consistency to determine someone's knowledge of some topic. [19] The concept is to ask questions to elicit someone's beliefs about the topic – if those beliefs are inconsistent, Aristotle would conclude that they didn't truly have said knowledge.

[1] I believe this came from Posterior Analytics but I can't find a direct source (SEP doesn't cite one).

2.2 Logic Programming

The following passage is from [11]. It is fairly concise, and so I thought worthy of being included verbatim:

> Arguably, Church's *Foundation of Logic* was the first Logic Programming language [3, 4]. It attempted to avoid the known logical paradoxes by using partial functions and disallowing proof by contradiction. The system was very powerful and flexible. Unfortunately, it was so powerful that it was inconsistent [12] and consequently the logic was removed leaving only the functional lambda calculus [5].

2.3 Computational Logic

2.3.1 Direct Logic

Gödel and Rosser both utilized self-referential sentences in their incompleteness proofs. Since these results, work has been done to develop restrictions which nullify these harmful sentences without taking too much power/expressibility away[2].

One research thrust in this direction was the development of Direct Logic [10], which only allows self-referential statements which are *Admissible*. Defined in [11],

A proposition Ψ is Admissible for a theory $\mathcal{T}$ if and only if

$$(\neg\Psi) \vdash_{\mathcal{T}} (\vdash_{\mathcal{T}} \neg\Psi)$$

This solved the problem caused by the classic self-reflective paradoxes i.e. the Liar and Russell paradoxes, but introduced a new one.

Namely, Direct Logic was used to (paraconsistently) prove the incompleteness theorem without assuming any form of consistency [11]. However, they discovered that Gödel's sentence was self-provable within Direct Logic, and hence every reflective theory in Direct Logic is inconsistent.

[2]This will be discussed in detail in Section 3.

2.3.2 Logic Theorist

Logic Theorist used substitution of axioms to find proofs of theorems. It is likely that, if it turned out that these axioms were inconsistent, that the system would be able to prove falsities (as was discovered of Naïve Set Theory with Russell's Paradox). However, as far as I can see, none of the authors of papers on Logic Theorist ever discussed this concern.

2.4 Automated Reasoning

Inconsistency is at the core of many forms of automated reasoning. Perhaps most significantly, resolution seeks to prove a sentence α is a consequence of a set of sentences Γ by proving that $\Gamma \cup \{\neg\alpha\}$ is inconsistent [16]. The idea to prove a sentence by refuting its negation was first discussed in [6], which describes what is now known as the Davis-Putnam algorithm.

2.5 AI

McCarthy's Advice Taker [14] is a problem solving program similar to the Logic Theorist and GPS. The driving motivation for the work was to create a machine which was imbued with common sense akin to that of humans.

Hewitt [11] stated that Advice Taker was found to be able to deduce inconsistent sentences. However this statement was not cited, and I cannot find any sources which corroborate or elaborate the point.

3 Inferential Inconsistency Relevant to Gödel-Rosser

3.1 Feferman 1984

Feferman [8] discusses possible solutions to the Liar Paradox by the way of restricting one of the following: (1) language, (2) logic, and (3) basic principles. In the context of the Liar Paradox and the Gödel-Rosser Theorem,

(1) consists of the ability to name statements and create self-referential statements.

(2) states that we accept the standard axioms and rules of propositional logic.

(3) that we have a truth predicate $T(x)$ which means that x is true, and hold the following axiom to be true: $T(\lceil\phi\rceil) \leftrightarrow \phi$.

The author argues that statement naming and self-reference are naturally abundant in natural language, and not inherently toxic. Consequently, we shouldn't resolve the Liar Paradox by removing either of these components. The author instead offers the Tarskian approach[3] in which the truth predicate T is only available in a meta-language. Formally, "Statement-naming $\lceil\phi\rceil$ is provided only for ϕ in L_0; thus $T(\lceil\phi\rceil)$ is a sentence of L only for ϕ in L_0." [8] However, this poses limitations on the logic which are discussed in detail in Restall 2006.

3.2 Restall 2006

Restall [18] discusses the notion of *non-classical* solutions to the Liar Paradox (and others). A non-classical solution is one which removes "problematic inferences" which enable paradoxes to be deduced. One such solution is that taken by paraconsistent logics: remove the inference rule $A \wedge \neg A \rightarrow B$. However, as one might expect, the author states that these solutions come at a cost:

> The general approach of using the paradoxes to restrict the class of allowable inferences places severe constraints on the domain of possible propositional logics, and on the kind of metatheory that is appropriate in the study of logic itself. [18]

[3]from his 1935 paper "The Concept of Truth in Formalized Languages", originally written in German.

4 Relationship between (In-)Consistency and ...

4.1 Defeasible Reasoning

As its name implies, Defeasible Reasoning is that which contains statements which can be *defeated* by new contradictory information. A defeasible argument is one that is rationally convincing but not valid purely by deduction [13].

One motivation for defeasible reasoning arises from a phenomenon in human reasoning. Humans often make assumptions based on their beliefs, consciously or unconsciously, which may later turn out to be false. Koons [13] provides two examples of this phenomenon:

> McCarthy found that we often make assumptions based on what is not said. So, for example, in a puzzle about safely crossing a river by canoe, we assume that there are no bridges or other means of conveyance available. Similarly, when using a database to store and convey information, the information that, for example, no flight is scheduled at a certain time is represented simply by not listing such a flight. [13]

These assumptions can aid our reasoning by filling in the gaps in our knowledge base. Of course, if a new fact is inconsistent with one of our assumptions, it needs to be removed, along with any deductions that were made on the basis of that assumption. Several systematic methods have been developed to remedy these inconsistencies, including Truth-Maintenance Systems and belief revision theory. Both of these are discussed further in their own sections.

4.1.1 Plan Construction

John Pollock argued in [15] that the Closed World Assumption is unrealistic in general reasoning scenarios, and because of this, plan construction must be defeasible. He created a refinement planner and implemented it in OSCAR, a cognitive architecture designed for general reasoning. The planner constructs a basic plan and modifies that plan when problems arise. It does this by tracking where beliefs and assumptions are used in the plan so that when a belief is defeasbily retracted, the parts of the plan which rely on that assumption can be revised.

4.2 Non-Monotonic Logic

Non-Monotonic Logic is typically defined negatively, as a logic whose consequence relation is not monotonic. Under a monotonic consequence relation, adding additional antecedents doesn't change the truth value of the consequence. For example, if we can prove the statement "The ground is wet." from the set of statements {"It has rained.", "It is not sunny."}, then we can still prove it from a larger set containing additional statements, e.g. {"It has rained.", "It is not sunny.", "The sky is green."}

In non-monotonic logic, this is not necessarily the case. Thus non-monotonic logics can be used for defeasible reasoning. However, it is more general and can be used for other types as well such as abductive reasoning.

Non-Monotonic Logic contains several tools for resolving inconsistencies. First, we differentiate between facts, which are known to be true, and defeasible statements, which can be defeated by contradictory information. If a fact and a defeasible statement are inconsistent, we simply keep the fact. [4]

A pair of inconsistent defeasible statements can be resolved in several ways. One is by considering the source of the statements in terms of reliability or authority, in the way that federal laws override state laws, which both override local laws.

The Specificity Principle states that when two inferences are inconsistent, the one with the more specific antecedent should be kept. The SEP article on Non-Monotonic Logic [20] describes an example involving penguins. Say we know the following sentences:

[4] If two facts are inconsistent I imagine we have a big problem...

$$\forall x\ (\texttt{Penguin}(x) \rightarrow \texttt{Bird}(x)) \tag{1}$$
$$\forall x\ (\texttt{Bird}(x) \rightarrow \texttt{Flies}(x)) \tag{2}$$
$$\forall x\ (\texttt{Penguin}(x) \rightarrow \neg\texttt{Flies}(x)) \tag{3}$$

We could infer that penguins can fly on the basis that all penguins are birds and all birds can fly. However this is inconsistent with our belief that penguins cannot fly. In accordance with reality, the Specificity Principle tells us to keep the inference with the more specific antecedent, "All Penguins cannot fly." than that with the more general antecedent, "All Birds can Fly and All Penguins are Birds". [5]

4.3 Truth-Maintenance Systems (TMSs)

Truth-Maintenance Systems (TMSs) manage and update a set of beliefs for a non-monotonic logical reasoner. Non-monotonicity is essential, as it allows a TMS to retract beliefs based on new information, which is, by definition, not possible in a monotonic logic.

In [7], Doyle calls a belief P "in" the current set of beliefs if "P has at least one currently acceptable reason" and "out" if "P has no currently acceptable reasons (either no reasons at all, or only unacceptable ones)".

A reason is an ordered pair of sets of beliefs, and is said to be *acceptable* if its first set of beliefs are all "in" and its second set of beliefs are all "out". [7]

Let

```
P = "There is not a flight from BOS to JFK today",
Q = "The list of flights includes a flight from BOS to JFK".
```

An acceptable reason for belief in P is $\{\{\}, \{\mathrm{Q}\}\}$. In words, we justify that there is not a flight from BOS to JFK because there isn't one on the

[5] Of course, we'd also rather revise (2) instead of removing it entirely, as a formal statement along the lines of "Most birds can fly." could still be useful in reasoning about other birds.

list of flights[6].

However, say this belief is updated – perhaps the list of flights is updated on the top of the hour, when a flight from BOS to JFK appears. Formally, let

```
R = "Flight 207, which leaves BOS for JFK at 1200 today,
is on the list of flights."
```

If we believe R to be true, $\{\{\mathrm{R}\}, \{\}\}$ is an acceptable reason for Q to be "in" the set of beliefs. That is, since we believe a specific flight is on the list, we believe there is a flight on the list in general. However, our reason for believing P is now unacceptable, because it relied on Q being out. Thus, we must remove our belief in P.

TMSs are also useful for formalizing defaults. Doyle [7] gives the example of a program for scheduling events and meetings. Say the program wants the default day for a meeting to be Wednesday (perhaps most meetings occur on Wednesdays). We formally justify the belief that a meeting is on Wednesday because we have no justification for the belief that the meeting is on any other day. When a justification arises (say, the belief that the meeting is on Tuesday), we discard the belief that the meeting is on Wednesday.

4.4 Belief Revision

Belief Revision is interested in maintaining a set of beliefs which is consistent, and methods to maintain that set's consistency when new, contradictory information arrives. The dominant theory is (currently) the AGM theory, which include three methods for belief change [9]. One of these, revision, is the process of adding a sentence to the belief set, and possibly removing other sentences so that the new set is consistent.

When a set of beliefs is inconsistent, there can be more than one choice of sentence which could be removed to restore consistency. AGM theory employs the concept of "entrenchment", in which all beliefs are given a value

[6] Or, at least, we have no evidence of a flight being on the list (perhaps we don't have access to the list yet).

based on their explanatory usefulness. To illustrate, imagine that your beliefs include the laws of thermodynamics and you observe what appears to be a perpetual motion machine. This observation leads you to believe that perpetual motion is possible. However, this contradicts your belief in the first and second laws of thermodynamics. Clearly you should throw away your belief in this supposed perpetual motion machine instead of your beliefs in the laws of thermodynamics.

As Atriya Sen mentioned during a presentation of this work on April 24, 2019, the concept of entrenchment could be very practical in scientific domains. However, it would be very challenging to create a measure of explanatory usefulness that would be effective in a general context.

4.5 (Computational) Paraconsistent Logic

In classic logic, one can infer anything from a contradiction. In many applications – especially automated reasoning, AI, and belief revision – this has the potential to cause huge problems.

In the "real world", inconsistencies are abundant. For instance, people often hold sets of beliefs which are inconsistent. We would like to have a logical system which can reason over inconsistent beliefs in a controlled, useful manner – that is, without allowing the agent to infer anything from a contradiction.

This is the goal of paraconsistent logics. In general, a paraconsistent logic is any logic which doesn't allow this "explosion" of inferring anything from a contradiction [17]. Specifically, paraconsistent logics don't include the inference rule $P \wedge \neg P \rightarrow \perp$.

The history of paraconsistent logic is quite interesting. Major developments were made in Brazil by Newton da Costa, and in Poland by Łukasiewicz and Jaśkowski, with neither group aware of the other's progress while their research was taking place [17]. Much of the early work concerned developing propositional calculi for paraconsistent logics which had useful properties. However I have found very little in the way of *computational* paraconsistent logics. Thus far, I have found two papers of note in this area.

[2] describes Paralog_e, "an extension of the ParaLog Logic Programming Language ... that allows direct handling of inconsistency." ParaLog itself is an extension of Prolog which incorporates paraconsistency. However, the original paper on ParaLog is written in Portuguese, and I have been unable to find an English translation. [2] details the syntax of ParaLog_e and shows how to create ParaLog_e programs.

[21] presents a paraconsistent logic and shows how Isabelle (an automated theorem prover for HOL) can be used to generate proofs in this logic. The authors have a few papers concerning automated theorem proving in paraconsistent logic, but they are the only papers I have found in this domain.

5 Conclusions

Back to the overarching drive for this research,

"Do any of these help solve The Problem?"

Maybe. From my reading thus far, it seems that these methods have only been applied in specific contexts which warrant their application. In other words, I haven't yet found any work which attempts to utilize several of these methods in a general system, in which the system would have to determine which method is appropriate in each situation (e.g. if two sentences conflict, should the Specificity Principle or Entrenchment be applied? Or should they each be considered, and calculate some final opinion based on their individual outputs?)

Also, while these fields of research all concern inconsistency, they don't all have the same goal. Belief revision seeks to regain consistency, while paraconsistent logic aims to reason intelligently with inconsistencies.

I think to truly solve "The Problem", some form of synergistic, general system is necessary, employing many or all of these methods (and possibly others).

6 Appendix: Some Reflections

I created this appendix to store ideas and reflections that I had while reading, that could potentially be useful or interesting for further discussion.

6.1 Multiple Defeasible Arguments

This may already exist,[7] but it would be interesting to consider a principle in Truth Management Systems in which those statements which can be proved from multiple distinct sets of defeasible arguments are considered stronger than those with only one set of arguments[8]. As an example, Godel's incompleteness theorem can be proved using Godel's sentence or Rosser's sentence, and under the assumption of ω-consistency or standard consistency. It seems to me that if there are multiple (non-trivially distinct) proofs of a statement – even if its arguments are defeasible – that statement is more likely to be true. Moreover, this principle could be useful in calculating the explanatory usefulness of statements in a defeasible logic for the purposes of using entrenchment as a method of inconsistency resolution.

6.2 Contradictions in Knowledge Bases

A contradiction in a knowledge base could indicate information is missing, and an AI agent which could explore its environment should possibly infer that it needs to further explore its environment in order to resolve inconsistency (i.e. pondering the problem without actively seeking new information will never get it anywhere).

E.g. Say agent A observes agent B pick up a light, and when B does the light turns on. A believes that picking up the light causes it to turn on. Later, B picks up the light and it does not turn on. A believes that picking

[7] I found a similar, but more basic, concept in Doyle 1979: a belief may have several justifications, and they consider a belief to be valid if at least one of its justifications is valid.

[8] Of course, we would also have to consider the relative strength of those defeasible arguments. Is three proofs based on fishy defeasible statements better than one proof based on more reliable ones?

up the light doesn't cause it to turn on. Contradiction.

However, if agent A was able to explore its environment, it may discover that agent B was pressing a button that A couldn't see, which was the true source of the light turning on, thereby resolving the contradictory inferences.

References

[1] Aristotle. Prior analytics. `http://classics.mit.edu/Aristotle/prior.html`, 350 B.C. Accessed: 2019-02-26.

[2] Bráulio Coelho Ávila, Jair Minoro Abe, and José Pacheco de Almeida Prado. Paralog e: A paraconsistent evidential logic programming language. In *Proceedings 17th International Conference of the Chilean Computer Science Society*, pages 2–8. IEEE, 1997.

[3] Alonzo Church. A set of postulates for the foundation of logic. *Annals of mathematics*, pages 346–366, 1932.

[4] Alonzo Church. A set of postulates for the foundation of logic. *Annals of mathematics*, pages 839–864, 1933.

[5] Alonzo Church. *The Calculi of Lambda-Conversation*. Princeton University Press, 1941.

[6] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM (JACM)*, 7(3):201–215, 1960.

[7] Jon Doyle. A truth maintenance system. *Artificial intelligence*, 12(3):231–272, 1979.

[8] Solomon Feferman. Toward useful type-free theories. i. *The Journal of Symbolic Logic*, 49(1):75–111, 1984.

[9] Sven Ove Hansson. Logic of belief revision. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edition, 2017.

[10] Carl Hewitt. Common sense for concurrency and strong paraconsistency using unstratified inference and reflection. 2008.

[11] Carl Hewitt. Development of logic programming: What went wrong, what was done about it, and what it might mean for the future. 2008.

[12] Stephen C Kleene and J Barkley Rosser. The inconsistency of certain formal logics. *Annals of Mathematics*, pages 630–636, 1935.

[13] Robert Koons. Defeasible reasoning. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edition, 2017.

[14] J McCarthy. Programs with common sense symposium on mechanization of thought processes. *National Physical Laboratory, Teddington, England*, 1958.

[15] John L Pollock. Oscar: An agent architecture based on defeasible reasoning. In *AAAI spring symposium: emotion, personality, and social behavior*, pages 55–60, 2008.

[16] Frederic Portoraro. Automated reasoning. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2019 edition, 2019.

[17] Graham Priest, Koji Tanaka, and Zach Weber. Paraconsistent logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2018 edition, 2018.

[18] Greg Restall. Curry's revenge: the costs of non-classical solutions to the paradoxes of self-reference. *Revenge of the Liar*, pages 262–271, 2007.

[19] Robin Smith. Aristotle's logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2018 edition, 2018.

[20] Christian Strasser and G. Aldo Antonelli. Non-monotonic logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2018 edition, 2018.

[21] Jørgen Villadsen and Anders Schlichtkrull. Formalizing a paraconsistent logic in the isabelle proof assistant. In *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXIV*, pages 92–122. Springer, 2017.