Piotr Kulicki

The Use of Axiomatic Rejection

1. Introduction

The central point of logic is an entailment relation, which in different contexts is expressed as an implication in a system, syntactic derivability or semantic entailment. Let α and β be two formulae, such that α entails β. Two fundamental ways of reasoning take their sources in this relation: modus ponendo ponens (MP) - having asserted α assert β, and modus tollendo tollens (MT) - having rejected β reject α. Both of them are well known, intuitively clear and extensively used in scientific, philosophical and common sense reasoning.

To build a formal system usually MP is used because we are interested basically in formulae which are valid. Such a system will be referred to as a positive system or positive part of a system. However there are some good reasons to formalise the opposite set of non-valid formulae – the negative part of a system. It was Aristotle who invented the idea of such a formalisation (see [8]), at the same time as he invented the axiomatic method for valid formulae. This way of formalisation is called axiomatic rejection. Later the technique of axiomatic rejection was developed by J. Łukasiewicz and his followers (see for example [8], [11], [12]).

In the system built using the technique of axiomatic rejection some formulae are rejected directly as rejected axioms, others indirectly with the use of rules of rejection. The basic rule of rejection is MT. Thus, because, the positive part of a system describes the entailment relation, which is used in a premise of MT, the negative part of a system cannot exist without its positive counterpart. In many cases other additional rules of rejection have to be used. They are specific for a particular logic, and express certain properties of it.

A simple example of such a presentation of logic can be given for classical propositonal calculus. The positive part is well known, and consists of a usual set of axioms and rules. The negative part of the system consists of the following single rejected axiom Ax^{-1} and the rule MT expressed by reversed modus ponens (MP^{-1}) and reversed Substitution ($Subst^{-1}$):

Ax^{-1} $\dashv p$

(Single propositional variable is rejected)

MP^{-1} $$\frac{\vdash \alpha \rightarrow \beta \,;\, \dashv \beta}{\dashv \alpha}$$

(If $\alpha \rightarrow \beta$ is provable and β is refutable, then α is refutable)

Subst^{-1} $$\frac{\dashv e(\alpha)}{\dashv \alpha}$$

(If a substitution instance of α is refutable, then α itself is refutable)

There are three main applications of axiomatic rejection:

(i) decidability proofs

(ii) completeness proofs

(iii) representation of negative knowledge in incomplete systems

In the following three paragraphs these applications will be briefly described and exemplified. This paper neither surveys the results on axiomatic rejection nor presents their details. The paper is meant to show the power of the method and variety of its use. The completeness issues have not received enough attention so far, thus showing its importance is the main novelty of this paper.

2. Decidability

The first application of axiomatic rejection - the use of axiomatic rejection for decidability proofs – is, relatively, best known. Classical logic as well as intuitionistic logic and most of the important modal systems received their axiomatic rejection presentation, which provide proofs of decidability of the systems (see for example [12]). It is possible to give such a proof also for systems lacking Finite Model Property, which is sometimes regarded as a necessary condition for decidability (see [11]).

There are two approaches towards decidability with the use of the method of axiomatic rejection. In the first of them it is proved that the set of theorems of a theory is decidable by showing that both sets of theorems and of non-theorems are recursively denumerable. These two sets are determined resp. by the positive and negative parts of an axiomatic system, and the fact that they are recursively denumerable can be simply observed from the finitary character of axioms and rules. Such a proof is valid but not necessarily useful for more practical purposes, such as building efficient decision procedures.

In the second approach, on the other hand, decidability is proved by showing an efficient decision procedure. In most cases a notion of a normal form, connected with the logic concerned, is used. Any formula can be transformed into an equivalent formula in a normal form in the positive part of the system. Thus, having the rule of modus tollens, it is enough to be able to reject all non-theorems in a normal form. A formula in a normal form can be decomposed to a set of simpler formulae, in such a way that it is a theorem if and only if at least one of them is a theorem. Thus if all formulae resulting from the decomposition are non-theorems, then the normal form is also a non-theorem. If we formulate that fact in a form of a rule of rejection, and add that rule to axiomatic formulation of rejection of simple formulae, we obtain the negative part of the system in concern. Normal forms, decomposition principles and simple formulae vary from system to system, but the general schema of decision procedure remains the same[1].

We will show both approaches in the example of the modal system S5. The first approach in this case can be based on the observation concerning the set of extensions of the system. By rejecting all its extensions we can obtain a negative axiomatisation of it. The set of all consistent extensions of S5 forms a chain of logics, and in each element of that chain necessity can be described by a finitely valued matrix. S5 is a limit of that chain, when the number of values goes to infinity (see [10]). In this situation, to reject all extensions of S5 the following rule of rejection can be used:

(S5-I) $$\frac{\dashv \alpha_1, \ldots, \dashv \alpha_n}{\dashv L\alpha_1 \vee \ldots \vee L\alpha_n}$$

where $\alpha_1, \ldots, \alpha_n$ are non-modal formulae. The rule can be interpreted as stating that there is no finite matrix for necessity in S5. Thus with any positive axiomatisation of theorems of S5 the axiomatisation of non-theorems consists of a single rejected axiom Ax^{-1} and the rules of rejection: MP^{-1}, $Subst^{-1}$ (taken from negative part of classical propositional calculus given above) and specific rule S5-I.

In the second approach the following S5 normal form can be used (see [1]):

[1] Presented way of building a decision procedure for a logic do not have to be connected with the axiomatic rejection. For example such a procedure is given by Scott for intuitionistic proposition calculus ([9]), by Lemmon for the modal system K ([7]) and by Hughes and Cresswell for S5 ([1]).

(*) $$L\alpha_0 \rightarrow L\alpha_1 \vee \ldots \vee L\alpha_{n-1} \vee \alpha_n$$

where $\alpha_0, \alpha_1, \ldots, \alpha_n$ are non-modal formulae. Formula (*) is a theorem of S5 if and only if at least one of the formulae $\alpha_0 \rightarrow \alpha_i$ $(1 \leq i \leq n)$ is. The following rule of rejection represents that fact.

(S5-II) $$\frac{\dashv \alpha_0 \rightarrow \alpha_1, \ldots, \dashv \alpha_0 \rightarrow \alpha_n}{\dashv L\alpha_0 \rightarrow L\alpha_1 \vee \ldots \vee L\alpha_{n-1} \vee \alpha_n},$$

The negative axiomatisation of S5 consists now of an axiom Ax^{-1}, and rules : MP^{-1}, $Subst^{-1}$ S5-II.

To decide if any formula α is a theorem of S5, we transform α into its normal form, decompose that normal form into simple formulae (i.e. formulae from the premise of the rule S5-II) and since they are non-modal check them in classical propositional calculus. If any of them is a theorem then it can be proved in S5 that α is also a theorem. If all of them are non-theorems, then they are rejected and so is α.

3. Completeness

The second use of axiomatic rejection is for showing completeness. It is more philosophically interesting than the first, more technical one, but its value hasn't been acknowledged so far. Its importance lays mainly in the possibility of obtaining completeness results without the use of formal models, in spite of the general opinion that completeness is necessarily connected with models.

Usually a formal system is built by stating true facts and correct rules to constitute a proof system. On the other hand, a formal model of a fragment of reality, which is formalised, is built. Then the proof system is confronted with the model and if they match it is regarded as sound and complete (see fig. I).

A model is a medium between the proof system and reality. Sometimes it is difficult to find a natural and clear formal model for a formalised fragment of reality and it is easier not to introduce it (compare [4]). Thus the possibility of considering completeness without model can be interesting and useful.

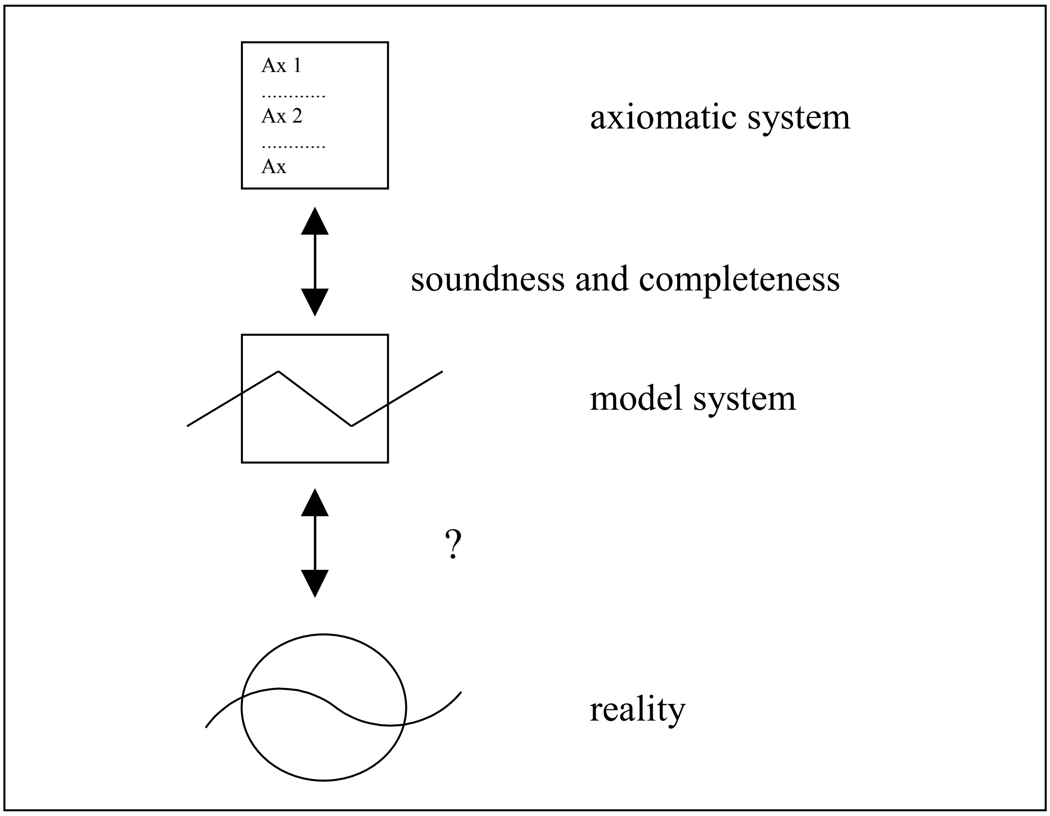

Fig. 1. Completeness considerations using formal model

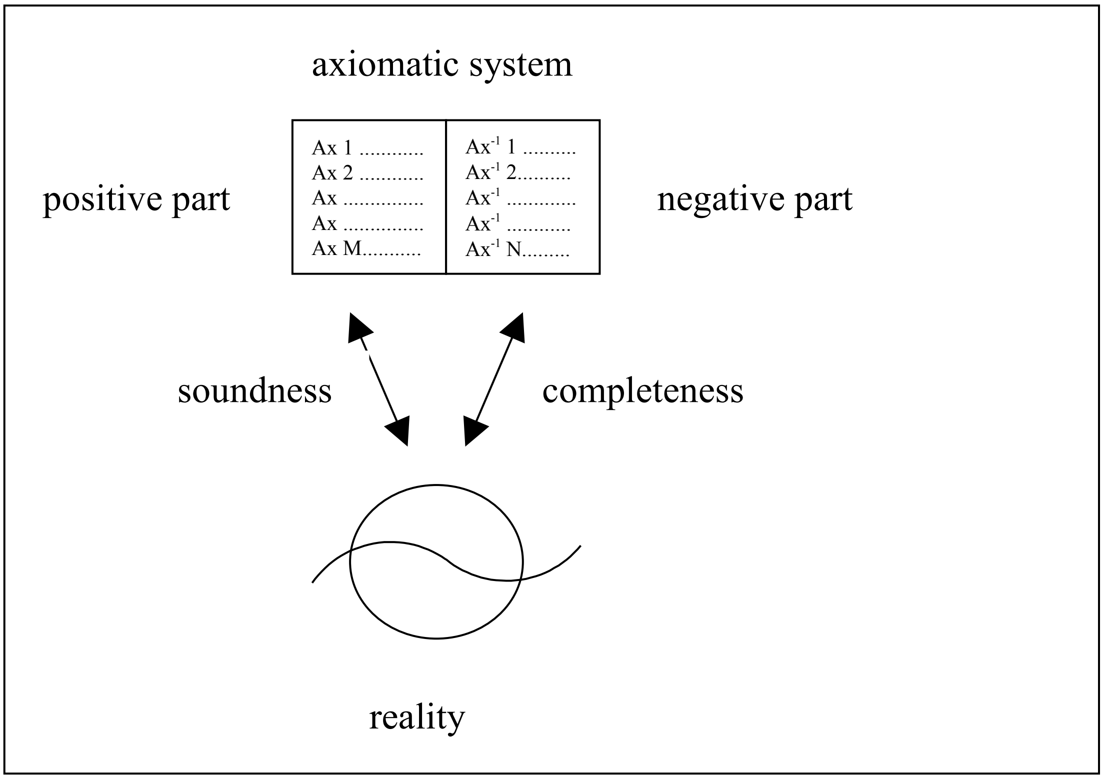

Fig. 2. Completeness considerations using axiomatic rejection

In general a system is complete if there is no valid statement of the considered language that is absent in the set of theorems. To show this it is enough to proof that all non-theorems are not valid. Using the technique of axiomatic rejection for such a proof we first build an axiomatic system of non-theorems. Then we turn from a purely syntactic presentation into semantics and prove that any rejected formula is not valid by showing that rejected axioms are not valid (the easiest way is by a counterexample) and that rules of rejection preserve non-validity (see fig.II).

Such a procedure will be described here using the example of different formalisations of Aristotle's syllogistic presented as an axiomatic first order theory. The first such formalisation was given by Łukasiewicz (see [8]). The language of the system consists of individual variables (X, Y, Z, ...), two binary predicates *a* and *i* (with infix notation) read respectively as *every ... is ...* and *some ... are ...* , and propositional connectives. As rules of derivation modus ponens, and substitution for individual variables are taken and the rules of rejection are reversed modus ponens (MP^{-1}), reversed substitution for individual variables ($Subst^{-1}$) and the following disjunction rule (Dysj):

(Dysj)

$$\frac{\dashv K \rightarrow A_1, \ldots, \dashv K \rightarrow A_n}{\dashv K \rightarrow A_1 \vee \ldots \vee A_n}$$

where $A_1, \ldots, A_n$ ($n \geq 1$) are atoms of the language and K is a conjunction of atoms. Axioms of the system, in addition to all substitutions of propositional calculus theorems, are as follows.

<u>System I axioms</u>

Ax1-I	XaX
Ax2-I	XiX
Ax3-I	$XaY \wedge YaZ \rightarrow XaZ$
Ax4-I	$XiY \wedge YaZ \rightarrow ZiX$
Ax^{-1}1-I	$XaZ \wedge YaZ \rightarrow XiY$

The rules of the system are not controversial. All except Dysj are straightforward and the rule Dysj can be interpreted as stating that the positive part of a theory can be axiomatized using Horn formulae - i.e. formulae of a form $A_1 \wedge \ldots \wedge A_n \rightarrow B$, where $A_1, \ldots, A_n$, B are atoms (see [6]), which seems to be right. However the set of axioms was criticized. Słupecki ([13]) noticed that Ax1-I and Ax2-I were never used by Aristotle. Indeed they are false when we take into consideration names without denotation like *unicorn* (Aristotle regarded all atomic statements with such names as false). Słupecki proposes another set of specific axioms with the same rules (rejected part was added later in [2]).

System II axioms

Ax1-II	$XaY \to XiY$
Ax2-II	$XiY \to YiX$
Ax3-II	$XaY \wedge YaZ \to XaZ$
Ax4-II	$XiY \wedge YaZ \to ZiX$
$Ax^{-1}1$-II	$XaY \wedge YaY \to XiX$
$Ax^{-1}2$-II	$YaX \wedge XaZ \wedge YaY \wedge ZaZ \to XaX$
$Ax^{-1}3$-II	$XaZ \wedge YaZ \wedge XaX \wedge YaY \wedge ZaZ \to XiY$

In this case we have the system that is sound (all positive axioms are valid) but not complete. With the natural interpretation of the predicates, axioms $Ax^{-1}1$-II and $Ax^{-1}2$-II are valid. Unlike the false statements *every unicorn is a unicorn* and *some unicorns are unicorns* similar statements with names with denotation like *every horse is a horse* and *some horses are horses* are true. The ancestors of both considered axioms provide that names used in predecessor have denotation. Thus we came up with another set of axioms which construct a system that is sound and complete.

System III axioms[2]

Ax1-III	$XiY \to YaY$
Ax2-III	$XaY \to XiY$
Ax3-III	$XaY \wedge YaZ \to XaZ$
Ax4-III	$XiY \wedge YaZ \to ZiX$
$Ax^{-1}1$-III	$XaZ \wedge YaZ \to XiY$
$Ax^{-1}2$-III	$XaX \to YiY$

[2] The proof that the negative part of this system is a complement of positive is similar to the proofs for previous axiomatisations.

Axiom Ax1-III represents the fact that for every name Y with denotation YaY is true. Axioms Ax2-III – Ax4-III are well known rights of Aristotle's syllogistic. For rejected axioms we can use the following counterexamples: in Ax $^{-1}$1-III we can put *horse* for X, *camel* for Y and *animal* for Z; in Ax $^{-1}$2-III – *horse* for X and *unicorn* for Y. This completes the proof of soundness and completeness of our axiomatisation of Aristotle's syllogistic.

In the proof only intuitions about single statements of the considered language were used. This is easier and safer method than introducing model, because there are several models that could be considered and choosing between them is more difficult then considering truth or falsity of individual statements. Furthermore syllogistic is a theory of two simple relations between names and is simpler then the notion of set and general concept of relation, which are engaged in formal models.

4. Negative information in incomplete systems

The third use of axiomatic rejection is connected with situations, when our knowledge is not complete. It gives a very natural way of formalising incomplete knowledge in computer programs written in Prolog. In the standard Prolog a negative answer is given for a query whenever there is no data for positive answer. With the incomplete information it can be misleading because the same answer is given when data allow for the justified negative answer for a query and where there is no data at all. That problem can be solved with the use of axiomatic rejection. A usual Prolog program is extended by a set of rejected formulae. If there is no positive answer for a query, then the query is added to the program and we try to derive rejected formulae in a program modified in this way. If we succeed, then by modus tollens we can reject the query and give the justified negative answer. Otherwise the answer is: *no data for answer.* [3]

As an example here is presented a program describing family relations in a situation of incomplete information (for details see [5]). Let it consists of the following clauses.

ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(X,Z), ancestor(Y,Z).
parent("John","Mark").
parent("Mark","Ann").

If we put a query about an entirely new person e.g. Jane, such as:

[3] Similar concept is present in [3] independently from the works on axiomatic rejection.

:- ancestor(“Jane”,”Ann”)

in standard Prolog we obtain answer “No”, as for a query for which negative answer is justified, like:

:- ancestor(“Ann”, “John”)

However if we add to the program a rejected formula:

ancestor(X,X)

and proceed as described the above two cases will be distinguished.

5. Conclusions

Various possible applications of the method of axiomatic rejection have been considered. It is useful for decidability proofs – even for systems without the finite model property, completeness proofs without the use of formal models and representation of negative information in incomplete systems. As an example of completeness result we have shown the correct axiomatisation of Aristotle’s syllogistic.

References

[1] Hughes, G.E., Cresswell, M.J., An Introduction to Modal Logic, Methuen, London, 1968

[2] Iwanuś, B., On Aristotle’s Syllogistic, Zeszyty Naukowe WSP w Opolu, Matematyka XXVIII (1972), pp. 41 – 56 (in Polish)

[3] Gabbay, D.M., Sergot, M.J., *Negation as Inconsistency*, Journal of Logic Programming, 1986, 1, 1-35.

[4] Kowalski R., Logic without Model Theory, in: D.M. Gabbay, editor. What is a Logical System, Oxford University Press, 1995.

[5] Kulicki, P., Logic of Programming and Aristotle’s Syllogistic, doctoral dissertation, Catholic University of Lublin, 1999 (in Polish)

[6] Kulicki, P., Computer Representation of the Contents of Genesis, Roczniki Filozoficzne, 1998, vol. I (in Polish)

[7] Lemmon, E.J., Scott, D., An Introduction to Modal Logic, Blackwell, Oxford, 1977

[8] Łukasiewicz, J., Aristotle's Syllogistic from the standpoint of modern formal logic, Clarendon Press, Oxford 1952

[9] Scott, D., Completeness Proofs for the Intuitionistic Sentential Calculus, in: Summaries of Talks Presented at the Summer Institute for Symbolic Logic, Cornell University Press, Ithaca, 1957, pp. 231-241

[10]Scroggs, S.J., Extensions of the Lewis System S5, Journal of Symbolic Logic 16 (1951), pp. 112 – 120

[11]Skura, T., Syntactic Refutations Against Finite Models in Modal Logic, Notre Dame Journal of Formal Logic, vol. 35, nr 4 (1994), pp. 595 - 605

[12]Skura, T., Axiomatic Refutation in Sentential Calculi, draft 1997

[13]Słupecki, J., Remarks on Aristotle's Syllogistic, Annales UMCS, vol. I, nr 3 (1946), pp. 187-191 (in Polish)

Piotr Kulicki

Katedra Podstaw Informatyki

Catholic University of Lublin

Al. Racławickie 14, 20-950 Lublin

E-mail: kulicki@kul.lublin.pl