



Repositorio Institucional de la Universidad Autónoma de Madrid

<https://repositorio.uam.es>

Esta es la **versión de autor** del artículo publicado en:
This is an **author produced version** of a paper published in:

Kybernetes 9.1 (1980): 37 – 44

DOI: <http://dx.doi.org/10.1108/eb005540>

Copyright: © 1980 MCB UP Ltd

El acceso a la versión del editor puede requerir la suscripción del recurso
Access to the published version may require subscription

AUTOMATIC SOLUTION OF SORITES

M. ALFONSECA

IBM Madrid Scientific Center, Paseo de la Castellana 4, Madrid (Spain)

A classical theory of syllogisms is shown that reduces to three-set theoretical inference rules, which have been used as the basis to produce a sorites (chain argument) solving program¹.

1 INTRODUCTION

Western medieval philosophy devoted a considerable effort to the development of a coherent body of knowledge which has come to be denoted as "classical logic", a prominent part of which is the theory of syllogisms. Modern symbolic logic on the other hand, has followed quite a different path, as a consequence of which classical logic has come to be an almost forgotten discipline, often restricted to high school curricula. While ample efforts are being exerted to produce a mathematical formalization of modern logic, we are still short of a comparable approach to classical logic.

2 CLASSICAL LOGIC

The elements of classical logic are the concepts, judgements and reasonings.

A concept can be considered as the mental representation of a set of objects. A term is the word expressing a concept; it is the name of the set represented by the concept.

A judgement is an assertion or negation linking two concepts. Its verbal representation is called a proposition.

Let A and B be two terms. They may be combined to generate the following 4 classes of judgements (and their corresponding propositions):

1. Universal affirmative, expressible by means of a proposition of the form: "Every A is B ".
2. Universal negative, "No A is B ".
3. Particular affirmative, "Some A are B ".
4. Particular negative, "Some A are not B ".

Two propositions are called equivalent if they express the same logical assertion or negation on the same pair of terms. Two different types of conversions generate equivalent propositions, namely:

- (a) Contradictory conversion.

* "Every A is B " is equivalent to "No not B is A ".

* "Some A are not B " is equivalent to "Some not B are A ".

This conversion is called contradictory because the structure of the phrase changes. In the first case, a universal affirmative proposition becomes a universal negative one. In the second case, a particular negative proposition is transformed into a particular positive one. In both cases, besides, one of the terms becomes negated by the process of conversion.

(b) Perfect conversion.

* "No A is B " is equivalent to "No B is A ".

* "Some A are B " is equivalent to "Some B are A ".

This conversion is called perfect, because the structure of the phrase in both equivalent propositions remains the same, the only difference being a transposition of their two terms.

Two propositions are called contradictory if they cannot be concurrently true or false. Two different cases occur:

* "Every A is B " versus "Some A are not B ".

* "No A is B " versus "Some A are B ".

Two propositions are called contrary if they can be false at the same time, but they cannot be concurrently true.

* "Every A is B " versus "Every A is **not** B ".

Two propositions are called subcontrary if they cannot be concurrently false. (They can be true at the same time).

"Some A are B " versus "Some A are not B ".

Two propositions P and Q are called subaltern if the following holds:

P is true $\rightarrow Q$ is true

Q is false $\rightarrow P$ is false

* "Every A is B " versus "Some A are B ".

A syllogism is the process of derivation of a new proposition out of two given propositions involving three different terms. As every proposition involves precisely two terms, one and only one of the three terms in the syllogism must appear twice, precisely once in each proposition. The solution of the syllogism, if any, is a new proposition involving the remaining two terms.

The two propositions of a syllogism are called its premises, while its solution is called the conclusion of the syllogism.

According to whether the premises of a given syllogism are affirmative or negative, universal or particular, and to the relative positions of the common term, different types of syllogisms can be considered. Classical logic has made a thorough study of all those combinations, with the following result: only nineteen different types of syllogism give rise to a correct conclusion, that is to say only 19 out of all possible combinations of premises, generate a correct proposition logically following from them. These nineteen types of syllogisms have received the following mnemonic latin names:

barbara, celarent, darii, ferio, cesare, camestres, festino, baroco, darapti, felapton, disamis, datisi, bocardo, ferison, bamalip, calemes, dimatis, fesapo, fresison.

A sorites, or chain argument, is a string of n propositions ($n > 2$), involving $n + 1$ different terms. Every term must appear exactly twice, in two different propositions, with the exception of two terms which must be mentioned only once, which are called the extreme terms. No two different terms may appear twice in the same pair of propositions. The solution of the sorites, if any, should involve the extreme terms.

Obviously, a syllogism is the particular case of a sorites when $n = 2$.

3 CLASSICAL LOGIC AND SET THEORY

If a concept can be considered as the mental representation of a set of objects, while a judgement is the assertion or negation of a relation between two objects. that is to say between two sets, it seems reasonable to assume that set theory should provide a good tool to the mathematical formalization of the theory of syllogisms.

In the following. we shall make use of the following symbols of set theory:

\subset	set inclusion
$=$	set equality
\neq	set inequality
A'	the complementary set of set A with respect to a given universe of discourse
Φ	the empty set

We shall also make use of the following symbols of the logic metalanguage:

\wedge	Logical "and"
\vee	Logical "or"
\sim	Logical "not"
\rightarrow	Implication
\leftrightarrow	Logical equivalence

3.1 Mathematical representation of judgements

Let us now try and express the judgement propositions of classical logic in terms of the preceding symbology. If we consider terms A and B as sets of objects, an assertion such as "Every A is B " can undoubtedly be rewritten as "Every member of set A is a member of set B ", and thus, would be equivalent to the expression

$$A \subset B$$

Classical logic usually assumes that an assertion of the form "Every A is B " implies the existence of A and B . For instance, if "Every man is mortal" were true, the existence of at least a man would be implied. In the set theoretical notation, this would be expressed as the assertion that the set of all men is not empty. Thus, in order to fully represent all the implications of the given proposition, we must expand the set theoretical expression in the following way:

$$(A \subset B) \wedge (A \neq \Phi)$$

Obviously, the preceding expression includes the assertion $B \neq \Phi$ as a corollary.

The proposition "No A is B " can be restated as "No member of set A is a member of set B ", or as "The set intersection of sets A and B is the empty set". This can be expressed in the following way:

$$A \subset B'$$

In this case, no assumption is made on the existence of either A or B . Thus, the proposition "No dragons are lazy" is considered to be true even when no dragons exist (the set of all dragons is empty). Thus, the preceding formula completely expresses the negation "No A is B ".

The proposition "Some A are B " can likewise be restated as "Some elements of A are

elements of B or "The set intersection of A and B is not empty", or "It is not true that *no A is B*". The latter being better fitted to our purpose, we shall express it with the following notation:

$$\sim A \subset B'$$

In a like manner, the proposition "Some A are not B " becomes:

$$\sim A \subset B$$

which can be phrased as "Not every A is B ", an obviously equivalent proposition.

In the last two cases, A is also supposed to be non-empty. However there is no need to include an explicit term in the mathematical expression, due to the fact that the proposition $\sim A \subset B$ logically implies $A \neq \emptyset$.

Summarizing the preceding discussion, we have come to the following results:

"Every A is B "	$\leftrightarrow (A \subset B) \wedge (A \neq \emptyset)$
"No A is B "	$\leftrightarrow A \subset B'$
"Some A are B "	$\leftrightarrow \sim A \subset B'$
"Some A are not B "	$\leftrightarrow \sim A \subset B$

Thus, universal and particular propositions respectively become assertions and negations on set inclusions.

3.2 Equivalent propositions

Two propositions should be called equivalent if they express the same relation of inclusion between the same sets. We shall now study the different types of equivalence recognized by classical logic, and the corresponding set theory equivalences they give rise to.

(a) Contradictory conversion.

* "Every A is B " is equivalent to "No not B is A " becomes:

$$(A \subset B) \leftrightarrow (B' \subset A')$$

* "Some A are not B " is equivalent to "Some not B are A " becomes:

$$(\sim A \subset B) \leftrightarrow (\sim B' \subset A')$$

(b) Perfect conversion.

* "No A is B " is equivalent to "No B is A " becomes:

$$(A \subset B') \leftrightarrow (B \subset A')$$

* "Some A are B " is equivalent to "Some B are A " becomes:

$$(\sim A \subset B') \leftrightarrow (\sim B \subset A')$$

Having in mind that

$$(P \leftrightarrow Q) \leftrightarrow ([\sim P] \leftrightarrow [\sim Q]),$$

where P and Q are any two propositions, is a tautology, and calling (if needed) B to B' (and thus B' to B), all four expressions above can be shown to reduce to the single well-known set theoretical formula

$$(A \subset B) \leftrightarrow (B' \subset A')$$

3.3 Opposed propositions

Let us now look at the different opposed propositions recognized by classical logic, according to the proposed set theoretical notation.

(a) Contradictory propositions.

* "Every A is B " versus "Some A are not B " becomes:

$$A \subset B \text{ versus } \sim A \subset B$$

* "No A is B " versus "Some A are B " becomes:

$$A \subset B' \text{ versus } \sim A \subset B'$$

Thus, contradiction becomes logical negation in both cases.

(b) Contrary propositions.

* "Every A is B " versus "Every A is not B " becomes:

$$\sim A \subset B' \text{ versus } \sim A \subset B$$

(c) Subcontrary propositions.

* "Some A are B " versus "Some A are not B " becomes:

$$\sim A \subset B' \text{ versus } \sim A \subset B$$

Both contrariness and subcontrariness change the second term by its complement with respect to the universe of discourse.

(d) Subaltern propositions.

* "Every A is B " versus "Some A are B " becomes:

$$A \subset B \text{ versus } \sim A \subset B'$$

Thus, subalternness becomes logical negation, plus set complementarization.

3.4 Syllogisms

Let $P(A, B)$ be a proposition defining a relation between terms A and B . A syllogism would be expressed thus:

$$P1(A, B) \wedge P2(A, C) \rightarrow P3(B, C)$$

$P1$ and $P2$ are the premises; $P3$ is the conclusion of the syllogism. If we consider all the possible combinations of the forms taken by $P1$ and $P2$, we would find the following cases:

(a) $P1$ and $P2$, are both universal propositions. There are only eight possible combinations, namely:

- | | |
|---------------------------------------|--|
| (1a) $A \subset B \wedge A \subset C$ | (1b) $A \subset B \wedge A' \subset C$ |
| (2a) $B \subset A \wedge A \subset C$ | (2b) $B \subset A \wedge A' \subset C$ |
| (3a) $A \subset B \wedge C \subset A$ | (3b) $A \subset B \wedge C \subset A'$ |
| (4a) $B \subset A \wedge C \subset A$ | (4b) $B \subset A \wedge C \subset A'$ |

All four cases labelled (b) become equivalent to those labelled (a) by means of the following conversion steps:

1. Convert the second premise into the one equivalent to it under a perfect conversion.

2. Replace C' for C and C for C' .

Example: case (1b) becomes:

1. $A \subset B \wedge C' \subset A$
2. $A \subset B \wedge C \subset A$ (case 3a).

Besides, case (3a) becomes the same as case (2a), according to the following conversion steps:

1. Change the order of both premises:

$$C \subset A \wedge A \subset B$$

2. Replace C for B and B for C :

$$B \subset A \wedge A \subset C$$

In conclusion: there are only three possible combinations of universal premises, namely those labelled (1a) (2a) and (4a) above.

(b) $P1$ is universal and $P2$ is particular. There are also eight possible combinations, namely:

- | | |
|--|---|
| (1c) $A \subset B \wedge \sim A \subset C$ | (1d) $A \subset B \wedge \sim A' \subset C$ |
|--|---|

$$(2c) B \subset A \wedge \sim A \subset C \quad (2d) B \subset A \wedge \sim A' \subset C$$

$$(3c) A \subset B \wedge \sim C \subset A \quad (3d) A \subset B \wedge \sim C \subset A'$$

$$(4c) B \subset A \wedge \sim C \subset A \quad (4d) B \subset A \wedge \sim C \subset A'$$

All four cases labelled (d) become equivalent to those labelled (c) by means of the same conversion steps as those converting cases (b) into cases (a).

In conclusion: there are only four possible combinations of one universal and one particular premise, namely those labelled (1c), (2c), (3c) and (4c) above.

(c) $P1$ is particular and $P2$ is universal. This case becomes the preceding one by a permutation of the premises.

(d) Both $P1$ and $P2$ are particular propositions. In this case no conclusion is possible.

We shall now try to find the respective conclusions in the remaining seven cases, aided by the rules of set theory.

* Case (1a) $A \subset B \wedge A \subset C$

If $A = \Phi$, nothing follows.

If $A \neq \Phi$, $B \cap C \neq \Phi$ follows. This is equivalent to $\sim B \subset C'$

Ergo:

$$A \subset B \wedge A \subset C \rightarrow \sim B \subset C' \quad (\text{RULE III})$$

* Case (2a)

$$B \subset A \wedge A \subset C \rightarrow B \subset C \quad (\text{obviously}) \quad (\text{RULE I})$$

* Case (4a) $B \subset A \wedge C \subset A$

Nothing follows.

* Case (1c)

$$A \subset B \wedge \sim A \subset C \rightarrow \sim B \subset C \quad (\text{obviously}) \quad (\text{RULE II})$$

* Case (2c) $B \subset A \wedge \sim A \subset C$

Nothing follows.

* Case (3c) $A \subset B \wedge \sim C \subset A$

Nothing follows.

* Case (4c)

$$B \subset A \wedge \sim C \subset A \rightarrow \sim C \subset B \text{ (obviously)}$$

If we apply contradictory conversion to the three propositions in the preceding relation, we get:

$$A' \subset B' \wedge \sim A' \subset C' \rightarrow \sim B' \subset C'$$

Replacing now A for A', B for B', C for C', we get:

$$A \subset B \wedge \sim A \subset C \rightarrow \sim B \subset C \text{ (again RULE II)}$$

and we have reduced case (4c) into case (1c).

The conclusion of the preceding discussion can be stated as follows:

All possible cases have been shown to reduce to only three rules of inference, namely:

$$\text{(RULE I)} \quad B \subset A \wedge A \subset C \rightarrow B \subset C$$

$$\text{(RULE II)} \quad A \subset B \wedge \sim A \subset C \rightarrow \sim B \subset C$$

$$\text{(RULE III)} \quad A \neq \Phi \wedge A \subset B \wedge A \subset C \rightarrow \sim B \subset C'$$

Example:

Every P is M

Some S are not M

Ergo: Some S are not P (Syllogism BAROCO)

$$P \subset M \wedge \sim S \subset M \leftrightarrow M' \subset P' \wedge \sim M' \subset S' \rightarrow$$

(RULE II) $\sim P' \subset S' \leftrightarrow \sim S \subset P$

4 IMPLEMENTATION

The preceding considerations have been used to implement an interactive sorites solving program. From the user's point of view, the program is invoked by typing SORITES at the terminal. A set of propositions are subsequently typed. They must be written in the English language, and together make up the sorites, the solution of which is desired.

The system may request further information to clarify the meaning of some of those propositions. Finally, when the user signals completion of the list of premises, the system types either the correct solution of the sorites, or a message stating its inability to find any conclusion.

The program accepts propositions built according to the following scheme:

Quantifier Subject Verb Predicate

The following terms are acceptable quantifiers: EVERY, NO, SOME, respectively corresponding to universal affirmative, universal negative, and particular propositions.

If the quantifier is omitted, EVERY is assumed.

The verb is limited to one of the two forms: IS, ARE. It may be followed by NOT, in which case, the proposition is taken to be negative. The verb is considered as a keyword separating the subject from the predicate.

The subject is that part of the proposition limited by the quantifier on the left, and the verb on the right. The predicate is that part of the proposition to the right of the verb. Both, are analyzed for negative particles, and denuded of those, final S's and articles.

A table is maintained by the program, the entries of which are all the subjects and predicates (subsequently called terms for brevity) in the propositions making up a given sorites. Once a term has been preprocessed, a table search is done to ascertain whether it has already appeared in previous propositions. The comparison algorithm may come to one of the following conclusions:

- a) Either an identical term is contained in the table.
- b) Or no similar term has been typed before.
- c) Or a similar term is found, in which case the user is requested to take a decision as to their equivalence.

In this way, typing errors and slight wording differences in the two appearances of the linking terms can be accounted for, and a valid conclusion may be generated by the program in those cases.

A proposition is finally converted by the analyzer into a vector of four numerical quantities, namely:

- a) The subject, represented by its index to the term table. If a negative particle was found by the analyzer within the subject, the sign of the index is negated.
- b) The predicate, similarly represented.
- c) A switch indicating whether the proposition is universal or particular.

d) A switch indicating whether the proposition is affirmative or negative.

Examples:

* Every human being is mortal becomes:

1 2 1 1

meaning that a universal affirmative proposition links entries numbers 1 and 2 of the term table.

* Some people who can not read are blind becomes:

-3 4 0 1

meaning that a particular affirmative proposition links entries numbers 3 and 4 of the term table. Entry 3 should be negated.

The term table would contain the following entries, once the analysis of the preceding propositions are complete:

Entry 1: Human being

Entry 2: Mortal

Entry 3: People who can read

Entry 4: Blind

The user may include comments (lines beginning by an asterisk) anywhere during the process of definition of the premises. These lines are ignored by the program.

Once the user has indicated the completion of the list of premises, the analysis program is exited, and control is transferred to the inference program, the data of which is the set of internal representations of all the premises, plus the term table. The program then tries to select successive pairs of propositions with a common term. A given pair may give rise to the following conditions:

a) One of the three inference rules is directly applicable. The program computes the conclusion of the pair, replaces both propositions by their solution in the list, and tries to select another pair.

b) One of the three inference rules is applicable if one or both premises are replaced by equivalent ones under a contradictory or perfect conversion. The program executes the replacement and goes back to step (a).

Let a proposition be internally represented by the quadruple

a b c d

The result of a contradictory conversion, according to the rules of set theory, can be shown to be:

-b -a c ~d

while the result of a perfect conversion is:

-b -a c d

c) No inference rule is applicable in any case. The program types the "no conclusion" message

d) The list of propositions gets reduced to a single quadruple.

A reconversion is then done, using the term table, into an English phrase, which is typed at the terminal as the conclusion of the sorites.

Once the program has found the solution (or the absence of a solution) of a sorites or syllogism, the user may decide either to type in a new chain argument, or finish the execution of the program.

The system is written in APL, and consists of 157 APL statements. A second version of the program accepts and analyzes Spanish phrases.

5 EXAMPLE

The following is an example of a session at the terminal. In the first place, the nineteen classical syllogism types are input to the system. In all cases, the conclusion exactly corresponds to the one expected, or to the result of applying a perfect conversion to it. The remaining examples have been taken from Carroll².

SORITES

* Examples taken from classical logic

* SYLLOGISM BARBARA *FELAPTON

Every M is P

No M is P

Every S is M

Every M is S

Ergo: Every S is P

Ergo: Some S are not P

*CELARENT

*DISAMIS

No M is P

Some M are P

Every S is M

Every M is S

Ergo: No S is P

Ergo: Some S are P

***DARII**

Every M is P
Some S are M

Ergo: Some P are S

***FERIO**

No M is P
Some S are M

Ergo: Some S are not P

***CESARE**

No P is M
Every S is M

Ergo: No P is S

***CAMESTRES**

Every P is M
No S is M

Ergo: No P is S

***FESTINO**

No P is M
Some S are M

Ergo: Some S are not P

***BAROCO**

Every P is M
Some S are not M

Ergo: Some S are not P

***DARAPTI**

Every M is P
Every M is S

Ergo: Some P are S

***FRESISON**

No P is M
Some M are S

Ergo: Some S are not P

***DATISI**

Every M is P
Some M are S

Ergo: Some P are S

***BOCARDO**

Some M are not P
Every M is S

Ergo: Some S are not P

***FERISON**

No M is P
Some M are S

Ergo: Some S are not P

***BAMALIP**

Every P is M
Every M is S

Ergo: Every P is S

***CALEMES**

Every P is M
No M is S

Ergo: No P is S

***DIMATIS**

Some P are M
Every M is S

Ergo: Some S are P

***FESAPO**

No P is M
Every M is S

Ergo: Some S are not P

*** EXAMPLES TAKEN FROM 'SYMBOLIC LOGIC', BY LEWIS CARROLL**

Every soldier is able to walk
Some children are not soldiers

No conclusion

Every soldier is a strong man
Every soldier is brave

Ergo: Some strong man are brave

Every well-fed lark is a powerful singer
No powerful singer is gloomy

Ergo: No well-fed lark is gloomy

Some dreams are terrible
No lamb is terrible

Ergo: Some dream are not lamb

Children are illogical people
No man who knows how to handle a crocodile is despised
Illogical people are despised

Ergo: No children is man who know how to handle crocodile

A buffalo is an animal able to push you through a wall
A donkey is unhorned
No unhorned animal is able to push you through a wall

With able to push you through wall, do you mean
Animal able to push you through wall?

Yes

With unhorned animal, do you mean
Unhorned?

Yes

Every not buffalo is a kicking animal

No phlegmatic animal is easy to swallow

With phlegmatic animal, do you mean
Kicking animal?

No

A kicking animal is a phlegmatic animal

Ergo: No donkey is easy to swallow

REFERENCES

1. M. Alfonsoeca, "Automatic solution of syllogisms and sorites" Current Topics in Cybernetics and Systems ed. J. Rose (WOGSC, and Springer-Verlag, Berlin, 1978) p. 36.

2. L. Carroll, El Juego de la Lgica (Alianza Editorial, Madrid, 1972).