Principles of Explosion and Its Way to Proof All Conclusions

Hafizh Afkar Makmur - 13514062[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13514062@std.stei.itb.ac.id

***Abstract*—Principle of Explosion is a principle that a contradiction can be used to prove anything. Two contradictory statements can be used to proof any conclusions even if the conclusion never even be mentioned in the premises. In this writing, the writer tries to prove that contradictory statements really prove all conclusions with various methods of proving conclusions.**

***Keywords*—Contradiction, contradictory premises, principle of explosion, proving conclusions**

I. INTRODUCTION

There are a lot of ways to build an argument to prove a point and convince the contender. One usually searches for premises that are considered true by both sided and try to show that the premises support his argument or decline his opponent's argument. A good process to find the premises and use it to attack opponent's arguments usually can bring a good victory to your side (unless the opponent rejects the process altogether and uses proof by intimidation). A mistake in using the process can result in more dangerous attack from the opponent to exploit the fallacy and maybe give him the chance to reject all your argument altogether (sometimes resulting in Fallacy Fallacy[1], which can give you a chance to fight back). One must be careful to handle the premises as it can give one glory or defeat.

What happens when the premises seems contradictory by itself? Principle of Explosion states that contradictory premises can be used to prove basically anything. It is parodied in this xkcd comic.

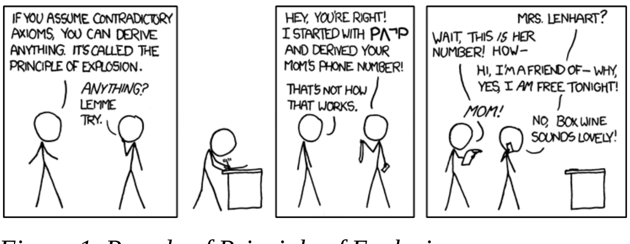


Figure 1. Parody of Principle of Explosion

In this writing, the author will try to prove that the Principle of Explosion really works with various proving methods. Of course the author will not try to derive anything from contradictory premises as shown in the comic, as it is not really the way Principle of Explosion works. But before touching the topic, we must know first what really is "proving conclusions".

II. USEFUL NOTES ON LOGICAL PROOF

A. Statement

One of the main compositions of every logical proof is a statement. Actually, logic can be called as knowledge of connecting statements [4]. A statement is defined as a sentence that has a definite true or false value. Another name of statement is *proposition*. Only a proposition can be used for a debate and become an argument just like an argument that "A wood is powered by the fiery fire of hell from the deepest ground of earth" or "Man did not land on the moon on 1969". There is no middle ground between whether the statement is true or not.

A statement sometimes is simplified to be an alphabet to be its symbol. It makes further proofing become easier and much simpler because now we do not handle a full sentence to refer for the original argument but only a single alphabet. A simple example is if I have arguments of "A man is better than a woman", "Windows is the greatest Operating System in 90's", and "Moon is actually a hole in the sky made by a divine when He tries to free himself from the mortal world and just the mortals perceive it as a celestial object", I can assign alphabet of p, q, and r for each of the argument. So, when I wanted to prove that "Moon is actually a hole in the sky made by a divine when He tries to free himself from the mortal world and just the mortals perceive it as a celestial object" I just have to proof that r is right, according to all other facts. It much helps when you handle a lot of statements and connections and especially if you do it hand-written, if reduce your effort for a same result.

B. Conjunction

Conjunction dictates relations between statements. There are a lot of types of conjunctions and few of them are 'and', 'or', 'xor' (exclusive or), and etc. These conjunctions connect statements and can be a useful tool to simplify some of the conversion of real life statements to be a logical statement.

[1] http://tvtropes.org/pmwiki/pmwiki.php/Main/FallacyFallacy

One of the examples is the statement “Marty is a cat and it is cute so if it bites you, you may not beat it or kick it.” You can assign a single alphabet for an entire statement (as usual according to the invisible convention of every logical book author, we use p). But, it is easier for further use if instead you identify every conjunction words and separate every clause to a different alphabet. For this sentence the substitution can be “Marty is a cat (p) and it is cute (q) but if it bites you (r), you may not beat it (s) or kick it (t)”.

To further beautify our compound statement, we use another symbol for every conjunction. There are many conventions for this and usually convention for hand-written symbol and computer symbol are different (Especially between different programming languages. One must remember the convention for the programming language every time one changes his language. Not that it is a difficult task.) For this writing we can use our own convention just for this case, ‘&’ for ‘and’, ‘|’ for ‘or’, ‘&’ for ‘so’ (‘but’ is considered to be the same as ‘and’), ~ for ‘not’, and ‘->’ for ‘if-then statement’ (it is a special case, there are many similar counterpart of this symbol to the natural language including word “so”, “necessary condition”, “sufficient condition”, and etc. It can be confusing to convert natural language of this type to logical symbol and *vice versa*). So our final compound statement can be formed like this

p & q & (r -> ~ (s | t)).

That is quite good than just a single p for that statement. This form is useful for further proofing and easier to analyze.

C. Values of Conjunction

Every conjunction has its own value of truth. Its value is influenced by its type, statement (or compound statement) in the left and statement (or compound statement) in the right. Its value dictates the value of the whole sentence, when usually value of the whole sentence is the value of the lowest-ranking conjunction in the sentence. The details of values of some of the conjunctions are listed below.

Conjunction ‘&’ is true if both of statements in its left and right are true. It is considered false even if only one of them is false (or worse, both of them are false).

Conjunction ‘|’ is true if just one of the statements is true. It can be true if only the statement on the right is true, the statement on the left is true, or both statements are true. It still can be false if both statements are false.

Conjunction ‘->’ is false only if the statement on the left is true when the statement on the right is false. Note that conjunction ‘->’ behaves a little differently than how the if-then statement usually behaves in natural language (or they are still both the same, if you treat the if-then statement in your language the same as that).

Conjunction ‘^’ or ‘exclusive or’ hold a unique nature. It is sometimes mistaken for or (‘|’) because in a lot of languages (including English, the language we’re talking in) ‘|’ and ‘^’ is both symbolized using the same word, ‘or’. ‘or’ that means ‘^’ actually appears rather seldom than ‘or’ that means ‘|’. One of the examples of or that means ‘^’ is or that is used in this sentence.

Do you prefer coffee <u>or</u> tea?

I want to buy 3 chairs <u>or</u> a sofa.

As you can see, if you answer the first question with “I want both! ”, it comes as a very rude statement, knowing that by the structure of the sentence it is implied that the host only wants to give either coffee OR tea, not both. It is the same with the second statement. It is a waste to buy 3 chairs AND a sofa together, either the treasurer will hate the one who request it, and the room will be too full of those.

Examples of ‘or’ that means ‘|’ are like this

You can buy our cards to give to your friends <u>or</u> your enemies.

Please buy the snack <u>or</u> the toy!

In those sentences, there’s no stopping anyone to buy <u>both</u>, ‘or’ in these sentences mean that the buyer can buy only one, but they can buy both of them too, and they will be very happy if the buyer really do it.

Then again, it’s all according to the context. Be wary whenever you try to convert any ‘or’ to a logical symbol.

C. Methods of Proofing

A statement is proven if it can be considered true by its connections with an already established fact. This fact can be a statement that is true or false by nature, by observations, or by derivation from another fact. These facts sometimes are called *premises*. Logical proofing actually is just manipulating premises that we have to be a conclusion that we want.

There is some proofing method that is being considered valid by scientific community, and some that the author will use here are the Truth of Table, Rule of Inference, Axiom Schemata, and Propositional Resolution.

D. The Truth Table

The Truth Table is a table that contains all possible values of any statements in the focus. It is useful to brute force our way to prove that a statement is always true, always false, or there is some cases that makes a statement is neither.

We can try to construct a truth table to proof an already established method. We can take Modus Ponnens to be our rabbit.

Modus Ponnens contains two premises, p -> q and q. It contains q as conclusion. By one glance, it seems that Modus Ponnens is just a common sense. Really, it seems so logical and to simple to be wrong. But don’t trust your eyes, we must try to distrust anything and prove it with our own hands.

To use the truth table, we must hold all premises to fill the table. We must fill the table with all combinations of what will be the value of the premises. We have premise

1, p -> q. To make it easier, along with p -> q, we must fill the table with all of its components, p and q. So the table will be just like this.

p	q	p -> q
T	T	T
T	F	F
F	T	T
F	F	T

For the second premise, since p is already on the table, we need to add the table no more. So we have already finished step one.

For the next step, we have two choices. There are two (from many) methods of proving with the truth table and that is proving by validity and proving by unsatisfiability. We will work first to prove by validity.

To prove by validity, we must construct a statement with the format of (premise1) & (premise2) & … & (premiseN) -> (conclusion). After that, the statement must be proven valid, or will be true in every condition it will ever be. For our case, the statement will be (p -> q) & (p) -> q. The truth table will be like this.

p	q	p -> q	(p -> q) & (p) -> q
T	T	T	T
T	F	F	T
F	T	T	T
F	F	T	T

The truth table show to us that the statement of (p -> q) & (p) -> q will be true in every condition. So, by this method it is proven that Modus Ponnens is correct and your eyes don't lie.

To prove by unsatsifiability, one must construct a statement with the format of (premise1) & (premise2) & … & (premiseN) & ~(conclusion). After that, the statement must be proven unsatisfiable, or will be false in every condition it will ever be. For our case, the statement will be (p -> q) & (p) & ~q. The truth table will be like this.

p	q	p -> q	(p -> q) & (p) & ~q
T	T	T	F
T	F	F	F
F	T	T	F
F	F	T	F

Once again it's proven that the statement will be false in every case it will ever be, and we can rest our case.

E. Rule of Inference

Rule of Inference simply is using one of a lot of rules that has been established to drive the premises to be a desired conclusion. Some of the rules are:

Modus Ponnens :

p -> q
p
____
∴ q

Modus Tollens

p -> q
~q
____
∴ ~p

Hypothetical Sylogism

p -> q
q -> r
____
∴ p -> r

And many more…

F. Axiom Schemata

Axiom schemata too use a lot of axioms (as in its name) and use it to derive conclusions. It also uses Modus Ponnens as one of its tools. The Axioms are

Implication Introduction:
$\phi \Rightarrow (\psi \Rightarrow \phi)$

Implication Distribution:
$(\phi \Rightarrow (\psi \Rightarrow \chi)) \Rightarrow ((\phi \Rightarrow \psi) \Rightarrow (\phi \Rightarrow \chi))$

Contradiction Realization:
$(\neg\psi \Rightarrow \phi) \Rightarrow ((\neg\psi \Rightarrow \neg\phi) \Rightarrow \psi)$
$(\psi \Rightarrow \phi) \Rightarrow ((\psi \Rightarrow \neg\phi) \Rightarrow \neg\psi)$

Equivalence:
$(\phi \Leftrightarrow \psi) \Rightarrow (\phi \Rightarrow \psi)$
$(\phi \Leftrightarrow \psi) \Rightarrow (\psi \Rightarrow \phi)$
$(\phi \Rightarrow \psi) \Rightarrow ((\psi \Rightarrow \phi) \Rightarrow (\phi \Leftrightarrow \psi))$

Other:
$(\phi \Leftarrow \psi) \Leftrightarrow (\psi \Rightarrow \phi)$
$(\phi \vee \psi) \Leftrightarrow (\neg\phi \Rightarrow \psi)$
$(\phi \wedge \psi) \Leftrightarrow \neg(\neg\phi \vee \neg\psi)$

F. Propositional Resolution

In this method, we have to change the premises to be in clausal form. Steps to change it is abbreviated as INDO or

I mplication out
N egation in
D istribution
O perators out

And after that we derive the conclusion from the clausal forms by using resolutions. Resolution principle in general is

$\{\phi 1, ..., \chi, ..., \phi m\}$
$\{\psi 1, ..., \neg\chi, ..., \psi n\}$
____
$\therefore \{\phi 1, ..., \phi m, \psi 1, ..., \psi n\}$

So, for every contradictory premise in two clausal forms, you can erase it as you combine the two clausal forms into a new clausal form. It doesn't mean that when you want to combine two clauses those have to have contradictory phrase. It can be but usually no one does it because it does not further the process to reach conclusion.

The conclusion is proven if we can reach empty clause from the resolutions.

III. LOGICAL PROOF OF PRINCIPLE OF EXPLOSION

A. Using the Truth Tables

We already have two premises by definition, and that's the only one we need. We suppose that the conclusion that we want to reach is q, so we can construct a truth table like this.

p	~p	q
T	F	T
T	F	F
F	T	T
F	T	F

And now we use the method of proofing by validity

p	~p	q	p & ~p -> q
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T

And we now can use the method of proofing by unsatisfiability.

p	~p	q	p & ~p & ~q
T	F	T	F
T	F	F	F
F	T	T	F
F	T	F	F

By those two methods we have already proven that q is proven by using those two premises. Its result should be obvious noting that p & ~p is a contradiction and thus should be true in whatever implication which they are on the left side, and make false every 'and' compound statement they are in. So, it makes no real consequences whatever statement q is, since p & ~p will make it provable in every argument they are in.

B. Using the Rule of Inference

We must use a little freedom that is given by a lot of rule of inference that exists. We must use it because there is no 'q' in the premise and yet we must have it exists in the conclusion. Here will be shown one of the ways to use the rules of inference to drive the premise to our wanted (any) conclusion

p

∴ p v q (Rule of Addition)

(p v q) <-> (~p -> q) (Material Implication)

~p -> q
~p

∴ q (Modus Ponnens)

As shown in the Truth Table proof, it is shown here that we can easily change q with r or s or (p -> q & r & r ^ ~s -> b) which we can substitute in the first step (Rule of Addition). After that, we can follow the flow and got our desired conclusion fresh from the oven.

C. Using Axiom Schemata

It was fun to use the axiom schemata to prove this conclusion, since just like in the Rule of Addition example above, we can summon a proponent out of nowhere from the first Axiom Schemata. Here is one of the ways it can be used like that.

p	(Premise)
p -> (q -> p)	(Implication Introduction)
q -> p	(Modus Ponnens)
(q -> ~p) -> q	(Contradiction Realization)
~p	(Premise)
~p -> (q -> ~p)	(Implication Introduction)
(q -> ~p)	(Modus Ponnens)
q	(Modus Ponnens)

Thanks to the ability of Axiom Schemata to "introduce" (as in "Implication Introduction") a proposition out of nowhere, we once again succeed to prove that our magical two premises can prove our desired conclusion without a hitch. Really, there is no stopping for those two.

D. Using Propositional Resolution

It is far easier to use the propositional resolution to prove this principle. If we have to brute force with the truth table, using 3 steps with the rules of inference, using 8 steps with the Axiom Schemata (yes it is fun, but by no means is it easy you know), we only have to use ONE STEP[2] in this method of proofing. You don't believe me? Watch as the author magnificently unfolds the way.

{p}	(Premise)
{~p}	(Premise)
{}	(Step 1: Resolution)

[2] Excluding steps to convert the propositions into clausal form. But if you see those are not really precious steps since there's no change at all when we use those steps.

As you can see, as of step 1 the proving process has already ended since we have already reached the empty clause, proving that the conclusion that is wanted to be reached (what conclusion?) is proven to be true. See, we didn't touch the conclusion **at all**. Propositional Resolution only states that once we have gathered all the clausal forms, we just have to proof that it is possible to reach an empty clause by resoluting the clauses that we already have. Since we don't have to use all clauses, we can list only the clauses that we need, thus in this example we only use the clauses from the premises. After all, it's enough, more than enough to proof the conclusion already.

It is the last method that we will try to proof that the Principle of Explosion worked. There are a lot of other ways to prove it the other way, which you can see in internet. In Wikipedia, you can see the name of other methods just like the proof-theoretic argument or the semantic argument, neither one will you understand (the lack of more understandable proof of this principle drive the author to write this).

IV. APPLICATIONS OF PRINCIPLE OF EXPLOSION

Principle of Explosion is one of the subjects in a section of logical studies with the name of Paraconsistent logic. Paraconsistent logic deals with contradictory statements in a discriminating way. Principle of Explosion is known as *ex contradictione sequitur quodlibet* or "from a contradiction, anything follows"[1].

Principle of Explosion proves that every premise that have contradictory premises will be useless since it can prove anything and by extension, proving nothing. One of the topics that use this is discussion of the Liar's paradox. It is also used in development of AI (Artificial Intellegences) since it's used to avoid any contradictory premises being used in the logical thinking to break all of the deciding process.

V. ACKNOWLEDGMENT

The first and foremost thanks from the author is to the God, the All-Seeing, All-Knowing, for giving the author strength and time to push his study to this step, and to pull all this tricks in one night, like he usually do.

Other thanks will the author give to Randall Munroe the owner of xkcd website which gives the author the first touch of this issue, inspiring the author to learn more and knowing new things related to this subject (and thanks for inspiring the author to search for another issue too, not just this).

Special acknowledgment belongs to the author's roommate that with his humility and good-will give the author a free room and a free time to free himself for any disturbance. If not for him, it's possible that the author will give up and just thinks that he can complete the task another day.

May the author succeed to pull his tricks once, twice, many times again, as he sails through the storm of tasks and tests, and once again prove that he is able to do it.

REFERENCES

[1] Carnielli, W. and Marcos, J. (2001) "Ex contradictione non sequitur quodlibet" Proc. 2nd Conf. on Reasoning and Logic (Bucharest, July 2000)

[2] Maulidevi, Nur Ulfa, Slide Perkuliahan IF2121 04-Propositional Proofs 2015, 11 Desember 2015.

[3] Maulidevi, Nur Ulfa, Slide Perkuliahan IF2121 04-Propositional Resolution 2015, 11 Desember 2015.

[4] Munir, Rinaldi, Slide Perkuliahan IF2120 Pengantar Logika (2013), 11 Desember 2015.

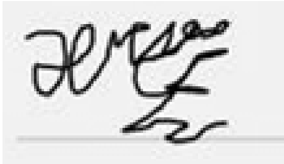
[5] Rosen, Kenneth H.
Discrete mathematics and its applications / Kenneth H. Rosen. — 7th ed. McGraw-Hill, 2007.

[6] https://xkcd.com/704/ accessed on 11 December 2015 on 1:30 AM

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2015



Hafizh Afkar Makmur - 13514062