Croatian Journal of Philosophy
Vol. XVIII, No. 54, 2018

Some Limitations on the Applications of Propositional Logic

EDI PAVLOVIĆ
University of Helsinki, Helsinki, Finland

This paper introduces a logic game which can be used to demonstrate the working of Boolean connectives. The simplicity of the system turns out to lead to some interesting meta-theoretical properties, which themselves carry a philosophical import. After introducing the system, we demonstrate an interesting feature of it—that it, while being an accurate model of propositional logic Booleans, does not contain any tautologies nor contradictions. This result allows us to make explicit a limitation of application of propositional logic to those sentences with relatively stable truth values.

Keywords: Logic gate, logic game, a priori, propositional logic.

1. *Introduction*

In this paper, we imagine first introducing (propositional) logic to a rational thinker via a simple logic game. It will be obvious this is a perfectly adequate way of explaining the Boolean connectives. At the same time, however, the bare-bones simplicity of the system will turn out to lead to some interesting meta theoretical properties, which themselves carry a philosophical import. That philosophical import has to do with the truths of logic, and the high rank these occupy among things we can know *a priori* (cf. Boghossian 2003, Harman 1996, Kitcher 1980, Peacocke 2005). It is useful to think that "[...] in general a logical truth is a statement which is true and remains true under all reinterpretations of its components other than the logical particles." (Quine 1961: 22–23). Sticking to propositional logic and the paradigmatic case of the law of excluded middle, $P \vee \neg P$, this idea would mean that this proposition's truth depends only on the logical symbols '$\vee$' and '$\neg$' or, in other words, is independent of what *P* might be (as long as it is a proposition). The properties of this system cast a doubt on that notion. In the following section of this paper, we will present the system in question, and its connection with propositional logic. In the central section, we demonstrate its meta-theoretical properties. Finally, the closing section of this paper takes a slightly bigger-picture perspective with an examination of the philosophical implications of the system.

2. *Lock-Key* game

The system at hand is presented in a form of a logic puzzle, one that asks whether a certain key opens a certain lock. Drawing inspiration from a computer infrastructure, locks are built of elements closely resembling logic gates, but with symbols altered to resemble the standard notation of Booleans in logic, including a specific orientation of the notation.

2.1 *Lock* elements

Lock consist of three kinds of elements, each with two subtypes: circles, squares and triangles, determined by the number of connections they have to other elements (one, two and three, respectively).

> *Definition 1* (*Circle*) Circle elements are the input/output elements of the lock. A number of *Input* circles are located at the bottom (i.e. "beginning") of the lock, and a single *Output* circle is located at the top (i.e. "end") of the lock. Note that a lock by definition contains only one output (and does in fact contain one).

Graphically we represent these elements as:

Input:
Output:

Fig. 1: Circle elements

> *Definition 2* (*Square*) Square elements have two connections—the one below considered its input and the one above considered its output. The types of this kind of an element are labeled *Same* and *Other*.

Graphically we represent these elements as:

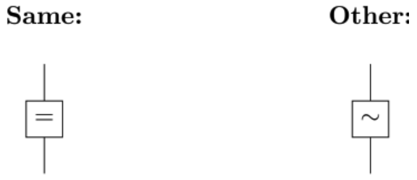

Fig. 2: Square elements

> *Definition 3* (*Triangle*) Triangle elements have three connections—the two below are considered its input and the one above considered its output. The types of this kind of an element are labeled *Minimum* and *Maximum*.

Graphically we represent these elements as:

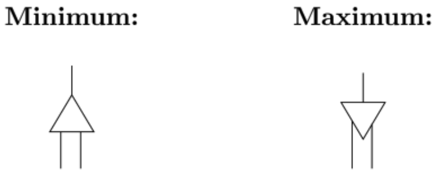

Fig. 3: Triangle elements

Note that Squares and Triangles are all *functions*, i.e. for any input they have only one output. The purpose of the Square element *Same* is purely for the sake of legibility, and will be discussed in the next section. The reader is probably aware of the purpose of the elements *Other*, *Minimum* and *Maximum*, but let us just note here that the shape of the latter two were chosen to facilitate memorization—*Minimum* is the widest at the bottom of the element, while *Maximum* is widest at the top.

We now proceed to introduce another element of the game—the keys, which serve as the primary input for the whole structure of a lock.

2.2 *Key* elements

> *Definition 4* (*Key*) A key is an ordered n-tuple $\langle v_1 \dots v_n \rangle$ where $v_i \in \{0,1\}$. A key is *a key to a lock* $\mathcal{L}$ just in case n is the number of input elements of the lock $\mathcal{L}$.

After introducing the elements, we now put them all together to describe how the "game" is played.

2.3 *Building a lock*

> *Definition 5* (*Lock building procedure*) A lock is built bottom up, and starts with a number of *Input* elements. We then add the other elements, with each of their inputs coming from an output of a lower element, and finally we add an *Output* element. The elements that some element ε gets its input from are called the *immediate predecessors* of ε.

Two limitations are that the output of each Square and Triangle is an input of some other element and that there is only one *Output* element.

In case we wish to consider only certain sections of a lock, we refer to them as *sublocks*:

> *Definition 6* (*Sublock*) Any element is a part of the same sublock as itself. If an element is part of a sublock, then the element(s) it gets its input from are also parts of the same sublock. We refer to sublocks as sublocks of their topmost element.

Example 7 An example of a correctly built lock is the following:

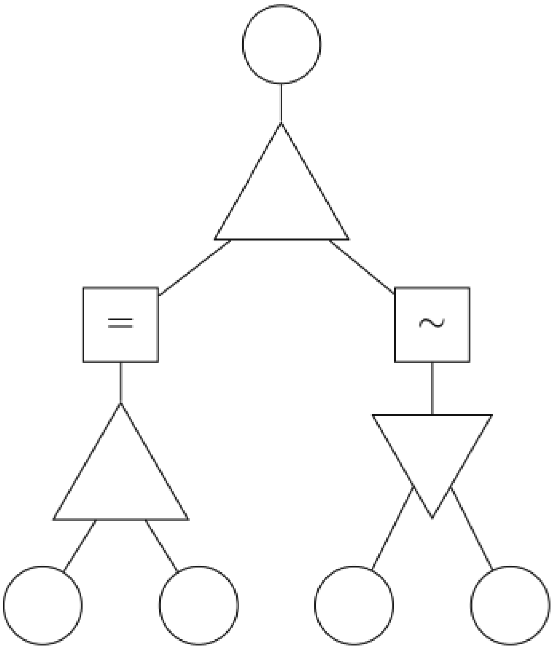

Fig. 4: An example of a lock

2.4 *Semantics of the game*

After introducing the lock and key elements, we now proceed to define their interactions:

> *Definition 8* (*Semantics*) The elements of a lock $\mathcal{L}$ produce the values as follows. Going from left to right, the *Input* element i takes the value v_i from the key to the lock $\mathcal{L}$. The element *Same* produces as its output the same value as it input, element *Other* produces the other value than its input, *Maximum* produces the greater of the two values in its input, and *Minimum* the lower. Finally, if the *Output* element takes the value 1 we take the lock to be "opened," and "closed" otherwise.

Example 9 Going back to the example lock from the Example 7, the only key to produce an "open" lock would be $\langle 1,1,0,0 \rangle$. It is left as an exercise to the reader to verify this.

2.4.1 *Connection to propositional logic*

It should be clear to the reader that

> *Observation 10* The semantics of the elements *Other*, *Maximum* and *Minimum* are precisely those of the Booleans *Not*, *Or* and *And*, respectively.

As an illustration, the Example 9 above would correspond to the formula $(A \wedge B) \wedge \neg(C \vee D)$ with the main connective being the conjunction, the left conjunct being a further conjunction, and the right one a negation of a disjunction. It is again left as an exercise to the reader to verify this formula is assigned 1 only when the variables *A, B, C* and *D* are assigned the values 1, 1, 0, 0, respectively.

Given this, it should be clear how this game will allow our thinker to understand propositional logic. We can also see why the element *Same* might be useful, as it reflects the idea that a subformula (which itself corresponds to a sublock) is assigned a value, but that value gets picked up by some other connective at a later stage. One can further utilize the game to introduce the semantic tables by considering how many possible keys there are for each lock. Note also that the shape of the triangle elements, while making sense in the context of *Minimum* and *Maximum*, also make it easier to distinguish the common disjunction and conjunction symbols.

We will now proceed to consider some metatheoretical properties of the game.

3. *Metatheory of the game*

As we have seen in the Example 7, all but one possible key produced a closed lock. The question that now arises is whether we can take this one step further by building a lock that is "unopenable."[1] Note that:

Observation 11 This question is equivalent to asking whether we can have a lock which all keys open, since we can transform one into the other by adding one *Other* element just before the *Output* element.

To encompass both of these, we will use the expression "forced."

Definition 12 (*Forcing a lock*) A lock can be *forced* if it can be made in such a way as to produce the same output for any possible key.

The answer to the former question is simply 'No' and the demonstration of this fact will be the central result of this paper.

Theorem 13 No lock in the *Lock-Key* game can be forced.

Proof. By showing no sublock can be forced, by induction on the last element.

Basic case. Input elements cannot be forced, by Definition 8.

Inductive case. If the last element is *Same*, its sublock can only be forced if the sublock of its immediate predecessor is forced. If the last element is *Other*, its sublock can likewise only be forced if the sublock of its immediate predecessor is forced, given Observation 11. Similarly for the Triangle elements, we can only force them if we can force the sublocks of their immediate predecessors.

Given Observation 10, it is clear that an unopenable lock would correspond to a contradiction, and a lock that every key opens to tautology.

In light of this, an interesting corollary follows:

Corollary 14 Lock-Key system contains no tautologies and no contradictions.

4. *Philosophical import*

What, if any, is the philosophical lesson and importance of this result? To see this, let us consider how to reconcile the apparent strain that exists between Observation 10 and Corollary 14. After all, the former tells us the elements of the game correspond to the Booleans, but those, in opposition to the corollary, do produce tautologies and contradictions (given our choice of the Booleans, the examples we'll focus on are $P \vee \neg P$ and $P \wedge \neg P$). The result that resolves the discrepancy, and is the main, if modest (given the limited scope of this paper) philosophical claim of this paper is the following:

Theorem 15 It is not the case that the tautologies [contradictions] of propositional logic are true [false] purely in virtue of the Booleans occurring in them.

[1] The author owes gratitude to Dragana Sekulić for bringing this question to their attention.

Proof. To see this, one should only note that the locks also contain *Input* elements, which correspond to the propositional variables. However, the semantics of these elements are not exactly alike—recalling the basic step of the proof of Theorem 13, *Input* elements cannot be forced. But the propositional variables can—the second occurrence of the variable P in either the example tautology or contradiction can have only one value—that of the first occurrence. This difference reveals an additional ingredient one needs to "cook up" a sentence of propositional logic that is always true or false—a certain type of behavior of the propositional variable(s).

Note that this argument does not show, nor does it in fact intend to do so, that the law of excluded middle (or the law of non-contradiction) is not true. Therefore, it is not a version of a "deviant logician" argument (cf. Williamson 2006, Sullivan 2014). The *Lock-Key* system corresponds to a system without a very basic, and by and large implicit, feature of propositional logic, that the propositional variables do not alter their value within an assignment. Such a system could be, e.g. one where we use a propositional variable only once in a formula. Alternatively, it could be one with such assignments which would allow the change of value while a formula is in use. This isn't entirely far-fetched, e.g. "I never wrote this sentence before." Rather, it just illustrates that logic is more regimented than we normally notice.

In explaining logic to our thinker using this game, we will have to explicitly and separately introduce limitations on the truth values of propositional variables. The very fact that the system is so basic allows it to make this feature of propositional logic apparent.

References

Boghossian, P. A. 2003. "Epistemic analyticity: A defense." *Grazer Philosophische Studien* 66 (1): 15–35.

Harman, G. 1996. "Analyticity regained?" *Noûs* 30 (3): 392–400.

Kitcher, P. 1980. "A priori knowledge." *The Philosophical Review* 89 (1): 3–23.

Peacocke, C. 2005. "The a priori." In F. Jackson and M. Smith (eds.). *The Oxford Handbook of Contemporary Philosophy*. Oxford: Oxford University Press.

Sullivan, A. 2014. "Logical deviance and the constitutive a priori." *Discusiones Filosóficas* 15: 67–85.

Quine, W. V. O. 1961. "Two Dogmas of Empiricism." In his *From a Logical Point of View*, 2nd ed. Harvard University Press.

Williamson, T. 2006. "Conceptual truth." *Aristotelian Society Supplementary Volume* 80 (1): 1–41.