Feedback and Paradox

Ferenc András

2023

Keywords: paradox, liar paradox, truth problem, finite automaton, logical circuit

Abstract

Every truth function corresponds to an isomorphic digital circuit. Consequently, the logical structure of every proposition can be presented within the range of propositional logic as an equivalent digital circuit. Provided that the logical values 'true' and 'false' correspond to the 'high' and 'low' voltage levels, the output of a circuit being equivalent with contradiction is always low level for every input state, whereas the output of a circuit corresponding to a tautology is always high level, irrespective of the input states. On the other hand, the remaining propositions correspond to circuits whose output is high-level if and only a few of the atomic components of the proposition are true, and the rest of the atomic sentence is false. The inputs equivalents to the atomic propositions are either high or low level. However, what logical circuits are equivalent to a circulating statement or argumentation?

The propositions are true or false irrespective of time, whereas the voltage level of the circuits can change over time. To be more precise, we can say that the input levels of the circuits are high or low depending on whether we valuate the atomic formulae of the formula expressing the logical structure of the proposition true or false. The voltage level of the output of the circuit and the truth value of the formula results from it. I will call 'combinational automaton' the digital circuit that may model the formulae of propositional logic. Formulae connected with truth functions yield formulae again. Although there are always corresponding automata for them, the situation is not simple in this case. We do not get combined automata joined to each other in each case, and it is also possible that we do not get an automaton—operating machine or circuit—at all.

There are digital circuits whose output is not a function of their input. The range of automata is wider than that of combinational automata. It includes machines whose input states do not determine unambiguously the output states, that is, the output is not a function of the input. It is because the circuit has feedback. Most digital circuits belong to this latter group. I will call them 'sequential automata'. The question arises whether there is a logical structure of circulating statement which corresponds to such a sequential automaton (or sequential circuit). In my view the logical structure of the Liar Paradox coincides with the operation of a sequential automaton irrespective of the logical correctness of the paradox itself. The analysis also examines possibilities of how the model could be further developed.

1 Finite state machines

Mealy automaton is a finite-state machine whose output values are determined parallel by the current inputs and its current internal state. A Mealy machine is a deterministic finite-state machine. In many cases, it is necessary to define the initial state of the automaton. It is presupposed that from some initial state, under the effect of some input signal, all states of the automaton are reachable. Some automata have no input, but their initial state is important. I call these automatic machines without input a generator.

- A deterministic finite-state machine is always in a unique, defined state and can only take a finite number of states. The automaton operates in discrete time. The atoms of time are sometimes called periods or moments. We use integers to represent atoms of discrete time. For any time t, the following time atom is $t + 1$, whereas the previous time atom is $t - 1$. Sometimes, I will refer to the present moment as t_0, the next moments as $t_1, t_2, \ldots$, and the previous moments as $t_{-1}, t_{-2} \ldots$.

- A is the set of internal states, X is the set of input states, and Y is the set of output states. We investigate the output state of the automaton that is uniquely determined by the internal and the input states. In our conception, Y is never an empty set.

- For a given internal state and an input or initial state, the automaton will step to the next internal state in the next time step. All possible state transitions are described by the function δ:following internal state = δ(present input state, present internal state)

- For a given internal state and an input or initial state, the automaton is set to a simultaneous output state. All possible output state options are described by the function λ:present output state = λ (present input state, present internal state)

These imply that the Mealy finite automaton is an ordered quintuple $\langle A, X, Y, \delta, \lambda \rangle$, where

$$a_1, a_2, a_3 \ldots \in A; x_1, x_2, x_3 \ldots \in X; y_1, y_2, y_3 \ldots \in Y$$
$$a_w = \delta(x_u, a_v); y_i = \lambda(x_u, a_v)$$

Hereafter, finite automaton is always understood to mean a Mealy-type finite automaton.

2 Combinational and sequential circuits

Every truth function corresponds to an isomorphic digital circuit. Consequently, the logical structure of every proposition can be presented within the range of propositional logic as an equivalent digital circuit. Provided that the logical values 'true' and 'false' correspond to the 'high' and 'low' voltage levels, the output of a circuit being equivalent with contradiction is always low level for every input state, whereas the output of a circuit corresponding to a tautology is always high level, irrespective of the input states. On the other hand, the remaining propositions correspond to circuits whose output is high-level if and only a few of the atomic components of the proposition are true, and the rest of the atomic sentence is false. The inputs equivalents to the atomic propositions are either high or low level. However, what logical circuits are equivalent to a circulating statement

First input	Second input	AND	NOR	XOR
0	0	0	1	0
1	0	0	0	1
0	1	0	0	1
1	1	1	0	0

Table 1: Logic gates

or argumentation?

The propositions are true or false irrespective of time, whereas the voltage level of the circuits can change over time. To be more precise, we can say that the input levels of the circuits are high or low depending on whether we valuate the atomic formulae of the formula expressing the logical structure of the proposition true or false. The voltage level of the output of the circuit and the truth value of the formula results from it. I will call 'combinational automaton' the digital circuit that may model the formulae of propositional calculus. Because the same atomic formulae can compose more complex formulae, one automaton can have more outputs, wherein every output corresponds to a complex formula.

Formulae connected with truth functions yield formulae again. Although there are always corresponding automata for them, the situation is not simple in this case. We do not get combined automata joined to each other in each case, and it is also possible that we do not get an automaton—operating machine or circuit—at all. To avoid this, it is sufficient to comply with the following rules:

(a) It is permitted to join any two inputs of any two automata.

(b) Arrange the automata on neighbouring levels and join the output of an automaton only with the input of another on a higher level.

At every point in time, the output state of a combinational automaton is unambiguously determined by the state of inputs. In other words, the output of the machine is the function of its input. The two rules given above will guarantee this. If these were not the case, it would not be possible to simulate the automaton by logical formulae because the truth value of a logical formula is unambiguously determined by the truth value of its atomic formulae; correspondingly, the truth value of a proposition is the function of the truth value of its components. Therefore, the rules given in (a) and (b) are interpreted in the world of propositions in the following way: the truth value of a proposition can never be influenced by its truth value (though it may be influenced by other attributes).

There are digital circuits whose output is not a function of their input. These are not combinational automata, but they can be constructed from components that are combinational automata themselves. They can be constructed for instance AND, NOR, and XOR gates.(Table 1)

The automaton below is based on AND, NOR, and XOR gates. $Input1$ and $Input2$ are the inputs of an XOR gate followed by two AND gates, and $Output1$ and $Output2$ are the outputs of two NOR gates. It operates in the following way: both input and output are either on a high level or low level at any point of time; the $output1$ is high or

*output*2 is low level at a point of time if *input*1 is low but *input*2 is high level. If the two inputs are on the same level, both high and both low, the output remains in the previous $(t-1)$ state. It means that the output is not a function of its inputs at the same time. Notice the feedback between *Output*1 and *Output*2. These automata are very important digital circuits. It has two inputs: $\langle input1, input2\rangle$ and two outputs: $\langle output1, output2\rangle$. The input variables are $\langle 00\rangle, \langle 01\rangle, \langle 10\rangle, \langle 11\rangle$; the output variables are $\langle 00\rangle, \langle 01\rangle, \langle 10\rangle, \langle 11\rangle$ ($\langle 10\rangle$ means $Output1 = 1$ and $Output2 = 0$). This machine is working in discrete time steps. Every time step period has a previous and next time step. The arrows show the direction of signals (Fig. 1).

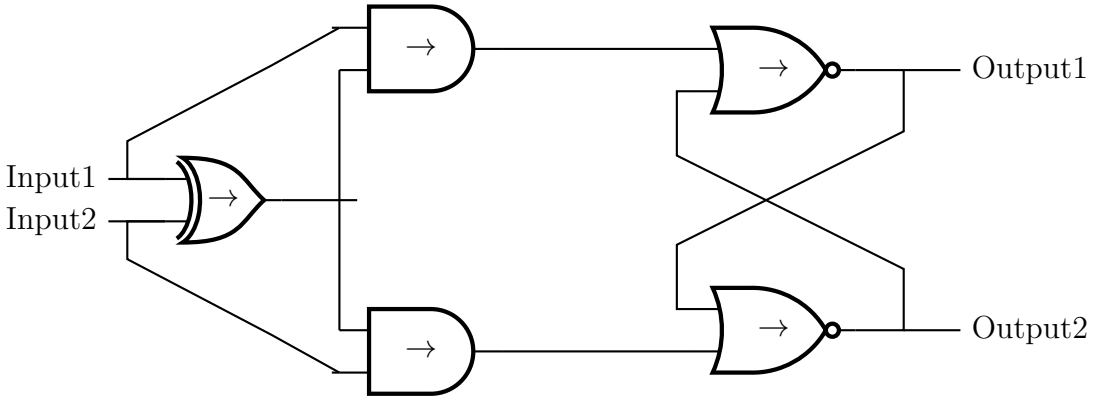

Figure 1: Flip-flop

The range of automata is wider than that of combinational automata. It includes machines whose input states do not determine unambiguously the output states, that is, the output is not a function of the input. It is because the circuit has feedback. Most digital circuits belong to this latter group. I will call them 'sequential automaton'. Figure 1 above is an illustration of such an automaton, but I will show other examples. [1]

δ	a	b
00	a	b
01	a	a
10	b	b
11	a	b

Table 2: Int.-state transition function

λ	a	b
00	10	01
01	10	10
10	01	01
11	10	01

Table 3: Output function

We can consider this circuit as a finite deterministic automaton (Mealey machine). The output state of finite automata depends on the previous internal state and inputs. More exactly, every event of outputs and internal status is determined by the previous events of inputs and previous internal status. We suppose the automata has an initial status, and it is $Ouput1 = 0$ and $output2 = 1$ and then every event of this automata is caused by its input events. The machine has two internal statuses: 'a' and 'b'. Where $a = 10$ means $Output1 = 1$, and $Output2 = 0$; $b =$ '$01'$ means $Output1 = 0$, and $Output2 = 1$. The automata remember the previous internal status 'a' or 'b'. Two functions—δ and

[1] In cyberspace, dead formulas come to life. They work in time; their operation can be tested. Please see `https://sht.andrasek.hu/flip-flop.xlsx`

λ—below describe the operation of automata. The left column of the tables is the set of input states, and the top row is the set of internal states (Table 2 and table 3).

following internal-state = δ (present internal-state, present input state)

present output state = λ (present internal state, present input state)

Logical relations between sentences may exist beyond propositional logic, corresponding to the operation of certain sequential logic circuits. What type of logic relationships can sequential circuits model? In the following paragraphs, I will provide examples of these.

3 The clock generator, as a model of the Liar paradox

The next digital circuit is simple, consisting of an inverter and a DELAY circuit. Notice the feedback again. The output of the inverter is high if the input is low, whereas the output level is low if the input level is high. We can consider the inverter as the model of the negation if the logical values 'true' and 'false' correspond to the 'high' and 'low' levels, respectively. For example, the input level corresponds to a valuation of the formula 'p', and the output level corresponds to the values of the formula '$\sim p$'. The unit named DELAY operates in a way that its output state in every $t + 1$ period is equivalent to the output state of the previous period t. Therefore, the automaton gives alternating high-level and low-level signals in the subsequent periods. The feedback from the output to the input simulates that the output level—the value of $\sim p$—is a function of itself. [2] To study the automaton, let us introduce the following notations:

$1 :=$ True

$0 :=$ False

$2 :=$ Not true and not false, i.e. value gap

$t + 1$:= the next period (time step) after t period

$\sim p$:= Not p

$|p|(t)$:= p formula valuation at t

$|\sim p|(t + 1)$:=$\sim p$ formula valuation at $t + 1$

Input := $|p|(t)$

Output := $|\sim p|(t + 1)$

Figure 2 shows the operation and the output signal of this machine.

The question arises again if there is a proposition corresponding to such an automaton. In my view, there is one to be found, though not a proposition but a circulating statement:

[2] This is how the electric bell works. The arm of the bell draws in only if there is an electric power in the coil, and conversely, electric power can run through the coil if the arm is not drowned in. In short, the bell arm only draws in if and only it does not draw in.

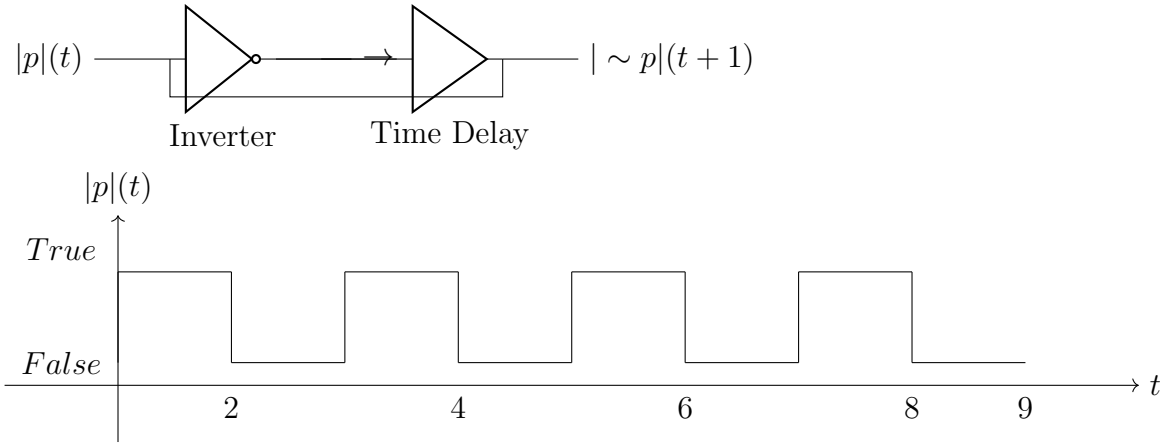

Figure 2: Liar paradox

the Liar paradox. The fact that the truth value of Liar sentence depends on itself is equivalent to the fact that the output of an automaton depends on itself because the output is fed back to the input. The Liar paradox has no constant truth assessment in time; we constantly conclude the opposite. Similarly, a feedback inverter does not have a constant level of output; it constantly switches to the opposite (the situation is different in three-valued logic that can also be modelled by an automaton). Hence, the logical structure of the Liar sentence coincides with the operation of the automaton, irrespective of the logical correctness of the paradox itself.

4 T schema

Let the object language sentence be that the 'The snow is white.' Let us introduce a notation for this: $p_0 :=$ The snow is white. The object language is denoted by L_0. The meta-language translation of the object language sentence is: $p_1 :=$The snow is white. In this case, the meta-language translation of the object language sentence is an identical mapping. The meta-language related to L_0 is denoted by L_1. The sentence p_0 in language L_0 is referred to as x in language L_1. Then, the following equivalences in L_1 are hold:

(T) x-is $True_1$ iff the snow is white; AND x-is $False_1$ iff the snow is not white.

In general form: $x = \ulcorner p_1 \urcorner$

(T) x is $True_1$ iff p_1; AND x is $False_1$ iff not-p_1.

The extension of the meta-linguistic predicates $True_1$ and $False_1$ consists of the names of the sentences of language L_0. For simplicity, I will refer to the meta-linguistic sentence p_1 as p in the following paragraphs.

The principles of bivalence are as follows:

$(B1)$ There is no sentence x, such that x is $True_1$, and x is $False_1$.

$(B2)$ There is no sentence x, such that x is neither $True_1$ nor $False_1$.

$(B3)$ x is not $True_1$ iff x-is $False_1$; AND x-is not $False_1$ iff x-is $True_1$.

$\{(B1), (B2)\} \Rightarrow (B3)$

5 Versions of Liar paradox interpretations

To emphasize the particularities of each interpretation, I will apply Kleene's strong three-value logic, and I simulate semantic valuations with finite automata operations.
Each version of the lying paradox is simulated by a clock generator, wherein the state of the generator corresponds to the truth value of the sentence. Each version differs in the operation of the internal state transition functions. Note the different tables of internal state transition functions.

5.1 Classical Liar paradox

$p :=$ This sentence is false.

Considering the sentence p, the sentence named x is false. A paradox arises if x is the name of sentence $p : x = \ulcorner p \urcorner$, and we do not use language levels. Applying the T schema above, if x is false, then not-p. Because $p = \ulcorner x$-is false.$\urcorner$, not-$\ulcorner x$-is false.$\urcorner$. Applying the principle of bivalence, therefore x-is true. Applying again the T schema, if x-is true, then p. But $p = \ulcorner x$-is false.$\urcorner$, so x is false. This brings us back to the initial point.[3]
If we suspend the principle of bivalence on the meta-language level, and if a sentence named x is neither true nor false because it is meaningless, then x is not false. However, sentence p says that x is false. These two are mutually exclusive, so if we accept that x is not false, we must again conclude that p is not corresponding to reality, that is, $\ulcorner p \urcorner$ is false. This brings us back to the initial point again, and it starts all over again.

The truth value of the sentence depends on itself, expressed by a feedback. The operation of the automaton simulates that no matter how we valuate sentence p, the automaton does not reach a constant state, alternating between true and false values.

The digital circuit model:

	In	Out	
	0	1	
$\lvert p\rvert(t)$	1	0	$\lvert \sim p\rvert(t+1)$
	2	2	
			Feedback

Table 4

Automaton model:

The operation does not depend on the initial state. The initial state can be 1 or 0 or 2, and the output will alternate between 1 and 0, and it repeats without end. This result is nontrivial and differs from the Kripke model or the strengthened Liar, regarding the different tables of the internal state transition functions.

[3]To see how the application of language levels blocks the paradox, see my paper "What is truth?". `https://ferenc.andrasek.hu/papers/what-is-truth6b.pdf`

δ	0	1	2
0	1	0	0
1	1	0	0
2	1	0	0

Table 5: Int. state transition function

λ	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

Table 6: Output function

5.2 The strengthened Liar

$p :=$ This sentence is false or has no truth value at all, because meaningless.

In this version, we suspend $(B2)$ on object language level. Consider the sentence p that the sentence named x is false or has no truth value at all, because meaningless. If sentence x is false, then it is the case as sentence p claims, so p is true. If the sentence x is true, then p is not true, because not correspond to reality, therefore p is false. If sentence x is neither true nor false because it is meaningless, then x is true, because it corresponds to reality, which implies that p is true. A paradox arises if x is the name of sentence p. Then the truth value of the sentence depends on itself, expressed by a feedback. The operation of the automaton simulates that no matter how we evaluate sentence p, the automaton does not reach a constant state, alternating between true and false values.

The digital circuit model:

	In	Out	
$\|p\|(t)$	0	1	$\| \sim p\|(t+1)$
	1	0	
	2	0	
			Feedback

Table 7

Automaton model:

δ	0	1	2
0	1	0	1
1	1	0	1
2	1	0	1

Table 8: Int. state transition function

λ	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

Table 9: Output function

The operation not depends on the initial state. The initial state can be 1 or 0 or 2, the output will alternate between 1 and 0, and it repeats without end. There is no fixed point. No matter what the initial state of the automaton is, that is, no matter whether the strengthened Liar sentence is considered true, false, or value-gap, the automaton does not reach a constant state. Note that δ function. The table describing the function differs from the Classical Liar Paradox (Table 5, 8).

5.3 Kripke's approach

p :='This sentence is false' sentence is ungrounded because it has no independent input that determines the truth value of the sentence.

The digital circuit model:

	In	Out	
	0	1	
$2 = \|p\|(t)$	1	0	$2 = \mid \sim p\|(t+1)$
	2	2	
			Feedback

Table 10

Automaton model:

δ	0	1	2
0	1	0	1
1	1	0	1
2	2	2	2

Table 11: Int. state transition function

λ	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

Table 12: Output function

The first row of the tables shows the internal states; the first column on the left shows the initial values. The operation depends on the initial state. If at time t, the value of p is 1 (i.e., true), then next time $t+1$ value is 0 (i.e., false), and vice versa. It repeats without end. The situation changes if the initial value is 2. If at time t, the value of p is 2 (i.e., value gap), then next time $t+1$ value also is 2 (i.e., value gap). The value gap is the fixed point of the function δ and λ.With this 2 (value gap) initial value, the automaton reaches a stable state. This result is characteristic of the Kripke model. Notice again the characteristic internal state transition functions that are different from previous versions of the paradox (Table 5, 8, 11).

5.4 Failed Liar

p:=This sentence is not true or false.

Considering the meta-language sentence p, the sentence x is not true or false, but meaningless. Whether a true or false sentence is denoted by x, it follows that p is false because p states the opposite. If x is the name of a meaningless sentence, then we must conclude that p is true because it asserts what is reality. However, this valuation is only valid in a temporary situation. For the particular situation that x is the name of the sentence p, it follows that p is true, which in turn is a negation of what p asserts; hence, p is false. Now, whatever initial valuation is chosen, the automaton reaches a constant state, and the constant state is 0.

The digital circuit model:

	In	Out	
	0	0	
$0 = \|p\|(t)$	1	0	$0 = \mid \sim p\|(t+1)$
	2	1	
			Feedback

Table 13

Automaton model:

δ	0	1	2
0	0	0	1
1	0	0	1
2	0	0	1

Table 14: Int. state transition function

λ	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

Table 15: Output function

Therefore, assuming that p sentence is true or false, the output of the automaton, independent of the internal state, is 0 (i.e., false). If we assume that the sentence is value-gap (i.e., 2), we get true in the first step and false in the next, and this is the constant state of the automaton. Thus, whichever way the sentence is evaluated, true, false, or value-gap, the output of the automaton is 0 (i.e., the sentence is false).

6 Other examples

6.1 The Truth-Teller is ungrounded

$p :=$ 'This sentence is true' sentence ungrounded because it has no independent input that determines the truth value of the sentence.

The digital circuit model:

	In	Out	
	0	0	
$1 = \|p\|(t)$	1	1	$1 = \|p\|(t+1)$
	2	2	
			Feedback

Table 16

Automaton model:

If the initial state is true, then the automaton reaches a true constant output state. If the initial state is false, then the automaton reaches a false constant output state. If we assume that the truth-teller sentence is meaningless, with no truth value, we get the value-gap output state. The automaton has three constant output states: true, false, or value-gap.

δ	0	1	2
0	0	1	0
1	0	1	0
2	0	1	2

Table 17: Int. state transition function

λ	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

Table 18: Output function

6.2 The Truth-Teller is either True or False

p:= 'This sentence is true' sentence either true or false.

The digital circuit model:

	In	Out	
	0	0	
$1 = \|p\|(t)$	1	1	$1 = \|p\|(t+1)$
	2	0	
			Feedback

Table 19

Automaton model:

δ	0	1	2
0	0	1	0
1	0	1	0
2	0	1	0

Table 20: Int. state transition function

λ	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

Table 21: Output function

If the initial state is true, then the automaton reaches a true constant output state. If the initial state is false, then the automaton reaches a false constant output state. If we assume that the truth-teller sentence is meaningless, with no truth value, finally we get a false output state too. The automaton has two constant output states, either true or false.[4]

7 Kripke's counter-example

7.1 Introduction to the problem

Saul Kripke refers to the following problem in his seminal study:[5]How do we evaluate the following statements from the point of view of truth, when Jones's statements ($s2$) and

[4]You can test the simulations in practice, in operation. Please see `https://sht.andrasek.hu/liar-versions4.xlsx`

[5]"Most (i.e., a majority) of Nixon's assertions about Watergate are false. ... Everything Jones says about Watergate is true." Saul Kripke, "Outline of a Theory of Truth", *The Journal of Philosophy*, Vol.72. No.19., (1975), p.691.

($s6$) and Nixon's statements ($s3$), ($s4$) and ($s5$) assert the following:

According to Jones:

($s2$) Most of Nixon's assertions about the Watergate scandal (more than half) are false.

In addition, Jones makes the assertion 'sentence6', which has the name ($s6$) and the truth value $|s6|$.

According to Nixon:
($s3$) All Jones's assertions about the Watergate scandal are true.

In addition, Nixon makes the assertions 'sentence4' and 'sentence5', the names of which are respectively (s4) and (s5), and the truth values of which are $|s4|$ and $|s5|$. The three independent assertions (sentences 4, 5 and 6) neither directly nor indirectly refer to assertions about the story – that is, they do not refer to either sentence ($s2$) or sentence ($s3$).

Let us determine which assertions were made independently by Jones and Nixon. It can be determined that Nixon made the assertions 'sentence4' and 'sentence5', while Jones made the assertion 'sentence6'.

$d2 = 1$:= Jones made the assertion ($s2$)

$e2 = 0$:= Nixon did not make the assertion ($s2$)

$d3 = 0$:= Jones did not make the assertion ($s3$)

$e3 = 1$:= Nixon made the assertion ($s3$)

...

Table 22 below shows who made which assertion:

Sentence-Person Relation	Jones	Nixon
($s2$) Most of Nixon's assertions about the Watergate scandal (more than half) are false.	$d2 = 1$	$e2 = 0$
($s3$) All of Jones's assertions about the Watergate scandal are true.	$d3 = 0$	$e3 = 1$
($s4$) sentence4	$d4 = 0$	$e4 = 1$
($s5$) sentence5	$d5 = 0$	$e5 = 1$
($s6$) sentence6	$d6 = 1$	$e6 = 0$

Table 22: Sentence-Person Relation

7.2 The basic valuation concept

The valuation of sentences ($s4$), ($s5$) and (s6) is an independent fact; it does not depend on either Jones's or Nixon's assertions. Note that both ($s2$) and ($s3$) use the 'true' or 'false' predicate, and that the truth of each sentence somehow depends on the truth of

the other. Based on this, the truth value of sentences $(s2)$ and $(s3)$ depends solely on the truth value of the sentences 'sentence4', 'sentence5' and 'sentence6', and on the mutual reference to the truth values $(|s2|)$ and $(|s3|)$. However, we can only freely evaluate the statements 'sentence4', 'sentence5' and 'sentence6', but not the sentences $(s2)$ and $(s3)$. The truth value of sentences $(s2)$ and $(s3)$ somehow follows from the valuations of the other three sentences, although we do not yet have a method for its precise determination. It is obvious that the truth values of $(s4)$, $(s5)$ and $(s6)$ determine whether the truth value of sentence $(s2)$ of Jones and sentence $(s3)$ of Nixon is true, false, or alternating. This means a total of eight variations, and we examine each of these cases below. The question then arises: In what language, and using what device, can we do this? How can we determine the truth value of Jones's assertion $(s2)$ and Nixon's assertion $(s3)$? Let us examine this more closely.

According to Nixon, every assertion made by Jones is true. Because Jones makes a total of two assertions, Nixon makes the following claim: Jones's assertions about the Watergate scandal are true if and only if most of Nixon's assertions (more than half) about the Watergate scandal are false and 'sentence6' is true. By using sentence notation, we are better able to see the logical connections/relationship.

Nixon: $(s3)$ is true if and only if $(s2)$ is true and $(s6)$ is true

Jones made a more complex statement about Nixon's assertions. According to him, most of Nixon's assertions are false. Nixon made three assertions. The majority of them can be false if and only if only one of the three is true. But what if not the majority, but all, of Nixon's assertions are false? What if Jones's sentence (s6) is false? How can all the options be calculated? In what language can the task be formulated so that we are easily able to calculate all the possibilities? Table 23 below summarises the above and presents the valuations used in the argumentation that follows. In the right-hand column, $e2 \times |s2| = 1$ if $|s2|$ is true and Jones makes the assertion s2; $e3 \times |s3| = 1$ if $|s3|$ is true and Jones makes assertion s3, etc.; the second line is $|s3| = 1$, precisely when all assertions made by Jones are true. The number of Nixon's assertions is $e9$.

Sentence	Valuation
(s2) Most of Nixon's assertions about the Watergate scandal (more than half) are false.	$\|s2\|$ = If $((0.5 > (e2 \times \|s2\| + e3 \times \|s3\| + e4 \times \|s4\| +$ $e5 \times \|s5\| + e6 \times \|s6\|)/e9)$; then $\|s2\| = 1$; otherwise $\|s2\| = 0$)
(s3) All of Jones's assertions about the Watergate scandal are true.	$\|s3\|$ = (If $d6 = 1$; then $\|s6\|$; otherwise 1) $\times$ (If $d5 = 1$; then $\|s5\|$; otherwise 1) $\times$ (If $d4 = 1$; then $\|s4\|$; otherwise 1) $\times$ (If $d3 = 1$; then $\|s3\|$; otherwise 1) $\times$ (If $d2 = 1$; then $\|s2\|$; otherwise 1)
(s4) sentence4	$\|s4\| = 0$ or 1
(s5) sentence5	$\|s5\| = 0$ or 1
(s6) sentence6	$\|s6\| = 0$ or 1

Table 23

7.3 Limitations to the classical truth theory

According to Alfred Tarski's basic insights, we have no guidelines for applying the concept of truth used by Jones at the first meta-language level, and the concept of truth used by Nixon at the higher, second meta-language level, or vice versa. It is obvious that, since each concept of the truth refers to the other, the two meta-language truth predicates cannot be on the same meta-language level. This is correctly pointed out by Kripke in his famous study. In the formal languages regulated by Tarski, the above argumentation about Jones's and Nixon's assertions, as part of the language, cannot be formulated applying some Tarskian concept of truth. This is correct for Kripke. (The other question is whether this limitation is a virtue or a mistake in a classical concept. I believe it to be a virtue.) On the other hand, neither Kripke nor anyone else who has developed an alternative truth concept is correct that, in the usual sense of the Tarskian concept of truth, the phenomenon itself, when we encounter semantic paradoxes, cannot be described. In this case, we perceive the paradox as a phenomenon, a series of events, and not as an argumentation in the framework of a logical language considered to be correct. In the following, I present models in which the behaviour of the models simulates the phenomenon when we encounter semantic paradoxes. Thus I do not construct another formal language, but rather models that describe the phenomenon on a meta-language level using semantic valuation functions, nevertheless within the framework of classical logic. The models simulate the semantic phenomenon. The models do not use the words 'true' and 'false', but the corresponding numerals '1' and '0'. In some cases, the numerals '1' and '0' are mere characters, otherwise they are numbers. The latter is the case when I use them as arithmetic operators.

7.4 Arithmetic transformation

$|s|$ denotes the truth value of any sentence s. The function ζ assigns numbers to the truth values: 0 for false, and 1 for true.

(4.1) $s2$ if and only if $0.5 > (\zeta|s3| + \zeta|s4| + \zeta|s5|)/3$

(4.2) $s3$ if and only if $s2$ and $s6$

(4.3) $s2$ if and only if $0.5 > (\zeta|s2| \times \zeta|s6| + \zeta|s4| + \zeta|s5|)/3$ (4.2)

Introducing the obvious definitions: $|s2| := \zeta|s2|$; $|s3| := \zeta|s3|$; $|s4| := \zeta|s4|$; $|s5| := \zeta|c5|$; $|s6| := \zeta|s6|$; this is what we obtain:

(4.4) $|s2| = \zeta|0.5 > (|s2| \times |s6| + |s4| + |s5|)/3)|$

Equation (4.4) is most easily solved using spreadsheet programs to obtain the following table of truths (Table 24). It should be borne in mind that $|s2|$ is the truth value of Jones's assertion, and $|s3|$ is the truth value of Nixon's assertion. The $|s3|$ value can easily be calculated by $|s2|$ and $|s6|$: $|s3| = |s2| \times |s6|$. Where there is no solution — no fixed point — there is no constant value in the spreadsheet, and the values fluctuate between 0 and 1.

sentence6	sentence5	sentence4	Jones's sentence	Nixon's sentence
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	There is no fixed point	There is no fixed point
1	1	0	There is no fixed point	There is no fixed point
1	1	1	0	0

Table 24

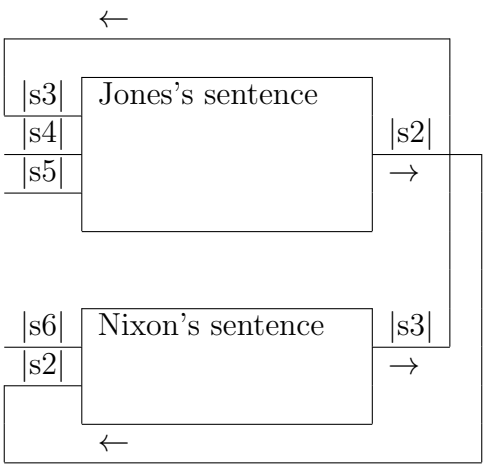

Table 25

7.5 Finite State Machine model

Table 25 illustrates how the logical connections between sentences can also be simulated with digital circuits. According to table 25, the output state of the circuits depends on themselves. Feedback simulates the semantical circularity of truth values. Such feedback circuits are called sequential networks because their operation cannot be described simply as a function of the input signals. The language of truth functions is not suitable to describe them, thus we need another mathematical tool.

Based on the above, we can define a Mealy finite automaton simulation of the problem. The output of the automaton is the value of Jones's and Nixon's sentences, its input is the value of the other sentences. The machine has two internal states that simulate the valuation situations of sentences.

(5.1) output = λ(input, internal state) – output function

(5.2) next internal state = δ(input, internal state) – state transition function

(5.3) internal state (situation) = $\{0, 1\}$

(5.4) input = $\langle |s6|, |s5|, |s4| \rangle$ – three independent sentences

(5.5) output = $\langle |s2|, |s3| \rangle$ – Jones's and Nixon's sentences

δ	0	1
000	0	1
001	0	1
010	0	1
011	0	1
100	0	0
101	1	0
110	1	0
111	0	1

λ	0	1
000	10	10
001	10	10
010	10	10
011	00	00
100	11	11
101	11	00
110	11	00
111	00	00

Table 26

The operation of the automaton—δ and λ function—is defined in Table 26.

The finite automaton model also works in cyberspace and generates output values for any input value. In the case of input where there is no solution to the problem – that is, where we get into a contradiction (no fixed point) then there is no stable (constant) state of the automaton and output alternates between 1 (true) and 0 (false). In this way, multiple versions of the liar paradox can be simulated using finite automatons. The simulation of semantic values by the finite automaton can be used in all cases where the domain of discourse is finite.[6]

8 Summary

Feedback logic circuits can simulate a significant fraction of semantic paradoxes (an exception is the Yablo paradox). Feedback simulates the self-dependence of truth values. We can consider such sequential automata as finite automata. Finite automata models can distinguish types of semantic self-reference of truth value. In doing so, they use the classical, Tarskian notion of truth. Their operation can be described by spreadsheet softwares and therefore the adequacy of the simulations can be verified in practice. This is because spreadsheet softwares can compute, unlike dead letters.

[6]The working model can be downloaded here:
`https://sht.andrasek.hu/kripke-s-counterexample.xlsx`