The Surprise-Test Paradox: A Formal Study

Vlad Vieru

E-mail: vlad@yukon.genie.uottawa.ca

Vlad Enache

E-mail: vlad.enache@connex.ro, venache@hades.ro

1. The Paradox

The teacher says: "Next week you will have a surprise-test."

The students reason: "If we will have the test on the last day of the next week (be it Sunday), then the previous day (Saturday) we could say: 'There was no test either on any of the previous days, or today, so it's for sure we will have the test tomorrow.' But a test that we know about a day before is not a surprise-test any more, so it's impossible to have a surprise-test on Sunday.

"Days Monday through Saturday are left. If we will have the test on Saturday, then Friday we could say: 'There was no test either on any of the previous days, or today, so it is for sure we will have the test tomorrow (Saturday) or the day after tomorrow (Sunday). But we have just established that a surprise-test is not possible on Sunday, so it is for sure we will have the test tomorrow.' But a test that we know about a day before is not a surprise-test any more, so it is impossible to have a surprise-test on Saturday.

"Days Monday through Friday are left. If we will have the test on Friday,..."

The students reach the conclusion that it is impossible to have a surprise-test on any day of the week. In other words, it is impossible for the teacher to keep his/her promise.

But if the test occurs on Wednesday, for example, there is no way the students could learn of it on Tuesday, so it is really a surprise-test, so the teacher keeps his/her promise.

2. Preliminaries: Clarifying the Terms

2.1 Logic

The paradox involves, among other things, the students' reasoning. Of course, we suppose the students think logically (we are not interested if they don't reason at all, or if they have firm ideas, or if they choose to toss up a coin to guess about the test). For studying their logical reasoning, we have to define a clear framework for expressing both their ideas and our ideas regarding the problems posed by the paradox. In the following, we set this framework to be the classical Predicate Calculus (PC).

A sentence is a word sequence that, as a whole, can be true or false. Sentences can be combined/prefixed by words like "and", "or", "not", "any...", "some..." etc., the results of such combinations being sentences, too.

A predicate is much like a sentence, but it has one or more undetermined words, called variables[1] (e.g. "Next week there will be a <x>"). When the variables are replaced by determined words (constants – e.g. "test"), the predicate generates a sentence ("Next week there will be a test"). Depending on the specific constants replacing the variables, the resulting sentence is true or false. In this study we will use capital letters (X, Y, Z, etc.) for predicates and lower case letters (x, y, z, etc.) for variables. The constants will usually be words in quotation marks ("Sunday", "test" etc.).

2.2 Time

Time is an essential ingredient of the paradox. By "time" we mean "the days of the next week" and their properties (e.g., one of the most important properties is that that they are ordered, which means the "days" come "one after the other"). This set of "days" ("the next week") defines the "temporal Universe", or the "time" of the problem. For brevity, we will use the short forms "day" and "week", but it should not be forgotten that they are not regular days or weeks, as we always talk in terms of "the time of the problem" (i.e., the "days" of that

[1] A sentence is a predicate with zero variables.

"next week" in which the teacher says there will be a "surprise-test").

All the properties of the "days of next week" can be built starting from the primary notions of "Sunday", "week" and "yesterday".

"Sunday" is an entity we consider as known, undoubtfully existing and which we are able to recognize and distinguish from other entities.

The "week" is a set of entities that we call "days" (to distinguish from other entities that are not part of the "week").

We suppose there exists a method of assigning each "day" some unique entity called "yesterday" (of that "day").

Given this, the "week" and its "days" have the following properties:

T1: "Sunday" is a "day" of the "week".

T2: For any "day" of the "week", "yesterday" is also a "day" of the "week".

T3: No two different "days" of the "week" have the same "yesterday".

T4: No "day" of the "week" has "Sunday" as its "yesterday".

T5: No matter how we try to build a "sub-week" (using "days" of the "week") so that it satisfies T1-T4, that "sub-week" will in fact be identical to the whole "week".

Notation: For an easier reference, "Sunday's" "yesterday" will be called "Saturday" and "Saturday's" "yesterday" will be called "Friday".

Remark 2.2.1 These are in fact Peano's axioms for Natural numbers. The analogy is as follows:

N	"week"
(natural) number	"day (of the week)"
zero	"Sunday"
successor	"yesterday"

The main difference is the meaning of "yesterday", which is not "next" (as for natural numbers), but "previous". But this is only an interpretation problem and it does not affect any theorem. This way, we can be sure any natural numbers theorem remains true if we interpret it in terms of "yesterday" and "Sunday" instead of "next" and "zero".

Remark 2.2.2 For the paradox to work, it seems essential that there is a certain "last day" (of "next week"). This property is captured by T1 and T4: T1 guarantees that at least one "day" exists (namely "Sunday"), while T4 guarantees that that "day" ("Sunday") is the last one (it is not "yesterday" for any other "day").

Remark 2.2.3 At first sight, the formal system of "the next week" built here is not working accordingly to the given paradox: the formal system generates an infinite number of "days", while it seems essential to the paradox to have just a finite number of "days".

In fact, nothing in the paradox requires a finite number of "days". For instance, "days" could really be all the intervals $[t_{n+1}, t_n)$, where $t_n = 1/n$, 1 is a constant arbitrarily chosen as unit and n is a non-zero natural number[2]. The "next week" will cover (0, 1), and at t=0 (which is not part of the "next week"!) the teacher could say "One of the days of the next week there will be a surprise-test".

The properties T1-T5 could be modified to model the "finite number of days" case. It is worth noting, though, that modeling the case of an infinite (i.e., "aleph-zero") number of days seems a more challenging endeavor, since it tries to attack a stronger version of the paradox.

Convention: The word (Anytime,...) in front of a sentence means (Any "day" of the "week", it is true that...). All the "today's" occurrences in the sentence name that "any day of the week".

---

[2] It can be proven that this set satisfies properties T1–T5.

Convention: The word (Sometime,...) in front of a sentence means (There is at least one "day" of the "week" when it is true that...). All the "today's" occurrences in the sentence name that "day of the week".

Convention: If there is no time setting for a sentence, we consider it speaks about all "days" of the "week". E.g.:

Students think rationally = Anytime, "today" students think rationally.

Remark 2.2.4 From T5 we can deduce the principle of complete induction:

If a predicate X satisfies the following two conditions:

I "Sunday" X

II Anytime,

("today" X) → ("yesterday" X),

then it is true that:

Anytime ("today" X).

2.3 Knowledge

An important role in the paradox is played by knowledge. "Knowledge" may mean "knowing for sure that", "believing that", "hoping that", "expecting that" etc. The "worst" case is when the paradox works for the most precise meaning – "knowing for sure that". That is why we will study this type of "knowledge".

In order to define it, it seems reasonable to accept these three generic properties:

K1: ("Today" "know" X) → X.

K2: [("Today" "know" X) and ("today" "know" Y)] ↔ ["today" "know" (X and Y)].

Remark 2.3.1 K1 says the same thing as "If X is false, then we do not know X." Or, yet: "We do not know any false sentence."

Remark 2.3.2 Though tempting, building K2' similar to K2, but with "or" instead of "and", would not be correct:

K2': [("Today" "know" X) or ("today" "know" Y)] ↔ ["today" "know" (X or Y)].

K2' is false!

K3: ("Today" "know" X) → ["today" "know" ("today" "know" X)].

Remark 2.3.3 This, coupled with K1, leads to the equivalence of knowledge and knowledge of knowledge:

("Know" Y) ↔ ["know" ("know" Y)].

We call this the "*meta-knowledge*" property.

2.4 Knowledge in Time

The properties of knowledge given so far are general properties, no time setting being involved. But for ensuring the consistency of knowledge over time, a new condition is necessary – that the students do not "forget" from one "day" to the other. This is a time-conditioned knowledge:

K4: ("Yesterday" "knew" X) → ("today" "know" X).

The paradox also assumes that, if the "test" did not happen on one "day", the students would know that (on the very same "day"). We also have to assume that if the "test" did happen on a certain "day", the students would know that fact (on that certain "day"). These assumptions lead to the following two axioms:

K5: ("Today" is "test") → ["today" "know" ("today" is "test")].

K6: ("Today" is not "test") → ["today" "know" ("today" is not "test")].

2.5 Surprise-test

The notion of "surprise-test" is not defined explicitly, but one thing is clearly stated:

["Yesterday" "knew" ("today" is "test")] → ("today" is not "surprise-test").

Common sense adds that if "today" there is no "test" at all, then it cannot be any "surprise-test" either: ("Today" is not "test") → ("today" is not "surprise-test").

So we know that:

[("Today" is not "test") or ("yesterday" "knew"

("today" is "test"))] → ["today" is not "surprise-test"].

In other words:

["Today" is "surprise-test"] → [("today" is "test") and ("yesterday" not "knew" ("today" is "test"))].

Other properties of the "surprise-test" do not occur, so (strictly for the necessities of this problem!) we may consider the above as the definition of "surprise-test":

S: ["Today" is "surprise-test"] ↔
[("today" is "test") and
("yesterday" not "knew" ("today" is "test"))].

Given this, we can write what the teacher says as follows:

P: Sometime, [("today" is "surprise-test").

Remark 2.5.1 We will read the teacher's "there will be a surprise-test" as "there will be exactly one test and it will surprise you". The paradox also holds for "there will be exactly one surprise-test" and "there will be at least one surprise-test", but these cases are mere technical complications of this case. For brevity, we will analyze the essential case only – "exactly one test and it will surprise you".

3. The Formal Framework

3.1 Notations

PC – the Predicate Calculus

P, P', P*, Q, Q1, X, Y ... – predicates/sentences

~ – logical negation (prefix)

or, and – the known logical operations

→ – logical implication

↔ – logical equivalence

Any, Some – the known logical quantifiers

of – "belongs to"

w – the next week

d, d', d*, d1, d2 ... – variable days (of w)

yd – yesterday of day d ("y" as prefix before the symbol of the day)

Sun – "Sunday"

Sat = ySun – "Saturday"

Fri = ySat – "Friday"

Kd(X) – "The students know on day d that X is true"

K(X) = Any d, Kd(X) – "The students know (anytime) that X is true"

Td – "A test happens on day d"

Sd = Td and ~Kyd(Td) – "A surprise-test happens on day d"

TBd – "A test happens before[3] or on day d"

SBd – "A surprise-test happens before or on day d"

3.2 Axioms

We consider the "Universe of the problem" is completely described by the following formal system (which we shall refer to as CFS – the *core formal system*)[4]:

PC = <the Predicate Calculus theory>

T1 = Sun of w

T2 = Any d of w, yd of w

T3 = Any d1, d2 of w,
(d1 ≠ d2) → (yd1 ≠ yd2)

T4 = Any d of w, yd ≠ Sun

T5 = Any w* ⊆ w, (w* satisfies T1-T4) → (w* = w)

K1 = K(X) → X

K2 = K(X and Y) →
[K(X) and K(Y)]

K3 = K(X) → [K(K(X))]

K4 = Any d of w,
Kyd(X) → Kd(X)

---

[3] "Before" is defined analogous to "greater than" for natural numbers.
[4] The inference rules in CFS are considered to be exactly the same inference rules holding for the Predicate Calculus theory.

K5 = T → [K(T)]

K6 = ~T → [K(~T)]

K7 = K(CFS)

U = Any d, d' of w,
[(Td and Td') → (d = d')]

Remark 3.2.1 K7 expresses the fact that the students know all the logical truths (axioms or theorems) holding in CFS. That is, the students can know anything which is given or logically deducible in CFS.

Remark 3.2.2 U represents an axiom of test uniqueness. It asserts that no more than one test (be it a surprise-test or not) is going to take place next week (K7 also ensures that the students know about that).

One could argue that U represents an artificial constraint. In real life, nothing prevents a professor from giving several tests "next week". However, we do not think that the essence of the paradox lies in the distinction between the cases "(at most) one test can happen" *versus* "any number of tests can happen". Provided the professor chooses to give more than one test next week, it can be shown not only that the students will be indeed surprised with *some* of the tests, but also a much stronger result: the students will be surprised with the very first test! Therefore, if one really wants to account for the "many possible tests" case, all he/she has to do is to reinterpret the predicate Td as having the following semantics "*The first* (and possibly only) test happens on day d" (see also previous Remark 2.5.1).

3.3 Two Important Sentences

We denote by P, respectively P*, the following two sentences:

P = Some d of w, Sd

P* = Some d of w, Td

Remark 3.3.1 P is the sentence uttered by professor P* is *implied* by P.

Remark 3.3.2 P is equivalent to SBSun, while P* is equivalent to TBSun.

Remark 3.3.3 Neither P nor P* are axioms in CFS. However, they *do* play an important role in CFS.

4. Useful Theorems

Let us show now several theorems provable in CFS.

First, two theorems "translated" from the Natural Numbers Theory, given here without their proof:

Theorem 1 = Any d of w, d ≠ yd.

Theorem 2 = Any d of w, (d ≠ Sun) → (some d' of w, d = yd')

Now, two results regarding knowledge acquisition in time:

Theorem 3 = Any d of w,
(d = Sun) or [Kd(X) → KSat(X)]

Proof: Let

Q(d) = (d = Sun) or [Kd(X) → KSat(X)]

be a predicate. We will use complete induction to show it stands for any "day":

Step I: Q(Sun) = (Sun = Sun) or [KSun(X) → KSat(X)]

is true by virtue of some basic laws in PC.

Step II: [Suppose] Q(d)

For d = Sun:

Q(yd) = Q(Sat) =
(Sat = Sun) or
[KSat(X) → KSat(X)]

is true by virtue of T1, Th1 and some basic laws in PC.

[PC says we may write]
For d = Sun,
Q(d) → Q(yd)

For any d of w, d ≠ Sun:

[Q(d)] (d = Sun) or
[Kd(X) → KSat(X)]

[PC allows us to drop the false parenthesis]
Kd(X) → KSat(X)

[K4] Kyd(X) → Kd(X)

[PC – "→" is transitive]
Kyd(X) → KSat(X)

[PC and T4]
(yd = Sun) or
[Kyd(X) → KSat(X)]

[Which is exactly] Q(yd)

[So] Any d of w,
d ≠ Sun, Q(d) → Q(yd)

[But we also showed that]
For d = Sun,
Q(d) → Q(yd)

[So] Any d of w, Q(d) → Q(yd)

[Through complete induction]
Any day of w, Q(d)

Any d of w,
(d = Sun) or [Kd(X) → KSat(X)]

Corollary 1 =
Any d of w, (d = Sun) or
[~Td → KSat(~Td)]

Proof: [K6] Any d of w, ~Td → Kd(~Td)

[PC allows this]
Any d of w, (d = Sun) or
[~Td → Kd(~Td)]

[Th3, put ~Td for X]
Any d of w, (d = Sun) or
[Kd(~Td) → KSat(~Td)]

[PC – "→" is transitive]
Any d of w, (d = Sun) or
[~Td → KSat(~Td)]

5. Analysis of the Paradox

5.1 The Bootstrapping Reasoning

The paradox suggests that a perfectly logical student is able to deduce there can be no surprise-test next week, and therefore the statement P uttered by the teacher is false. First, the student's reasoning seems to imply the impossibility of a surprise-test being given on Sunday. Since the series of deductions leading to the impossibility of a Sunday surprise-test seems to "bootstrap" the whole reasoning of the student, we shall conventionally call it the *bootstrapping reasoning*.

The sketch of the bootstrapping reasoning, as made by the student himself, is as follows:

"Let me suppose the test is given on Sunday. In this case, I would be able to know on Saturday, that, since no test has been given so far, the test must be given on Sunday. No surprise-test can be given, therefore, on Sunday."

It is crucial to realize that the above phrase is just a shortcut. The bootstrapping reasoning is, in fact, sensibly more complex. Let us make explicit its main steps:

"Let me suppose a test is given on Sunday. In this case, no test has been given on Saturday or before. **[Step 1]**

Moreover: if no test is given on Saturday or before, I know that on Saturday. **[Step 2]**

So if the test is given on Sunday, I know on Saturday that no test has been given so far. **[Step 3]**

But I also know [now and on Saturday] the structure of time – for example, I know that Sunday is the last day of the week, I know that Saturday is Sunday's yesterday and I also know that all the other days come before Saturday. Thus if I know on Saturday that no test has been given so far, then I am able to know on Saturday that the test will be given on Sunday. **[Step 4]**

So if the test is given on Sunday, I am able to know that on Saturday. **[Step 5]**

But to know on Saturday about the test being given on Sunday means the Sunday test is not a surprise-test. **[Step 6]**

To conclude: no surprise-test can be given on Sunday. **[Step 7]**"

5.2 Formalization

Let us formally translate the student's reasoning:

TSun → ~TBSat	(1)	(see axiom U)
~TBSat → KSat(~TBSat)	(2)	(provable using Corollary 1 in Section 4)

TSun → KSat(~TBSat)	(3)	(implication transitivity)
KSat(~TBSat) → KSat(TSun)	(4)	**(???)**

TSun → KSat(TSun)	(5)	(implication transitivity)
KSat(TSun) → ~SSun	(6)	(using the definition of S)

~SSun	(7)	(implication transitivity)

If looked upon in a hurry, the reasoning above may *appear* to be valid. The transitivity of logical implication is the only inference rule being used, and it is a valid rule. What about the conditional clauses themselves?

(6) can easily be proved, being a consequence of the surprise-test definition. However (5) is itself obtainable from (3) and (4). (3) is true, since it is obtained from (1) and (2), which are true.

Provided (4) was true, (5) and (7) would also be true. But is it (4) true?

As a matter of fact, (4) is ***not*** true. In order to assert (4), it is not enough to know the structure of time. Knowledge about the structure of time ensures only something weaker than (4):

if a student knows on Saturday that the test has not been given so far **and if he knows that a test is to be given at all**, then the student is able to know on Saturday that a test is going to be given on Sunday.

Or, otherwise stated, knowledge about the time structure entitles the student only to the following deduction:

"Provided I know (on Saturday) a test is to be given at all, then from knowing on Saturday that the test has not been previously given, I may also know on Saturday that the test is given on Sunday"

This is formally expressed by (4'):

KSat(TBSun) →
[KSat(~TBSat) → KSat(TSun)] (4')

As shown, (4) can be inferred from (4') by *modus ponens* only if the following also holds:

KSat(TBSun)

Or, using the previously defined notations:

KSat(P*) (8)

Indeed, the student may conclude on Saturday that a test will be given on Sunday *only* if he knows on Saturday that a test *must* be given next week. But does he really know that on Saturday?

Before we provide an answer, let us first remark that it seems far more natural to discuss about knowing *today* (and all the other days) P* rather than about knowing P* only *on Saturday*.

While K(P*) → KSat(P*) is true, the reverse implication is not true.

What would it be to know P* on Saturday, but not today? If a test is given on Saturday (or before), then learning TBSun exactly on the test day seems natural. However, in order to validate the student's reasoning, we need to have (8) true also for the case when no test is given on Saturday or before. In this case, knowing P* on Saturday, but not before, seems completely unsupported by intuition, let apart the logical formalism. How could the student possibly learn on Saturday that a test *must* be given if he did not know it before? To admit that the student *can* somehow learn such a thing eventually means to acknowledge that the student has a revelation exactly on Saturday. But while it may be a conceivable (although very controversial) rationale behind K(P*) – the student knows today P* because he hears the professor uttering P*, it seems there is no reason – at least, no *explainable* reason – to justify [KSat(P*) and ~Kd(P*)]. The professor

will not say anything new on Saturday, so he cannot "induce" sudden knowledge about the test occurrence only on Saturday.

For all the above reasons, we shall discuss the truth of K(P*), not just of KSat(P*). With this mention, the main question becomes really this: **does the student know in advance that the professor is actually going to give a test next week?**

Or, if we put it formally: is K(P*) true?

The answer is straightforward: neither K(P*) nor K(~P*) is provable in the formal framework we set so far (CFS), and the same goes also for P and P* (along with their negations).

In the following Subsections, we study what would happen if CFS is enhanced by one or more axioms.

5.3 Adding ~P* to CFS

Suppose we add to the system CFS the following axiom: ~P*

Since P* is just a notation for TBSun, the new system (CFS + ~P*)[5] represents a model for a (possible) world where, despite the professor's statement, there is not going to be any test next week.

In this case, ~K(P*) would be a theorem (since one cannot know something false) and the student's reasoning is clearly unsound, since it makes use of K(P*), which is false.

Furthermore, there are two choices:

5.3.1 Adding ~K(~P*) to (CFS + ~P*)

In such a system, the student does not know there is going to be no test next week. However, not even the perfectly logical student can discover this. He cannot conclude even the fact that there is not going to be any *surprise-test*. What is more, the student will be uncertain about the possibility of a Sunday test even on Saturday night. The student is doomed to experience uncertainty about the test until Sunday.

[5] Here, the semantics of "+"is understood to be: the axiom ~P* is added to CFS. Similar conventions will apply from now on.

5.3.2 Adding K(~P*) to (CFS + ~P*)

In this case, the student *does* know from the very beginning that there is not going to be any kind of test next week. Therefore, no reasoning about a surprise-test would make much sense.

Remark 5.3.2 CFS + ~P* + K(~P*) is actually equivalent to CFS + K(~P*)

5.4 Adding P* to CFS

Adding P* to the axioms of CFS means the professor is definitely going to give a test (be it a surprise-test or not) next week. This fact is now set as a formal truth.

But while *we* can take the truth of P* for granted in our study, the student may not know P*. (After all, he is confronted with a real-world situation, not with a formal system.)[6]

Since a true fact may be known or may be unknown, there are two possibilities:

5.4.1 Adding ~K(P*) to (CFS + P*)

This seems to be the model which best describes the situation outlined by the paradox: the professor is going to give a test next week, but the student does not know that in advance. Since the student relies on a false hypothesis ("K(P*) *does* hold"), his whole reasoning is unsound. Since the bootstrapping reasoning is flawed, the student cannot soundly conclude even the impossibility of a surprise-test on Sunday.

On the other hand, no matter on what day the test is given next week, this is going to be a genuinely surprise-test. Suppose the test is not given before Sunday. Then, no matter what the student *believes*, the fact is that he does not know even on Saturday whether a test is going to be given or not on Sunday. Since he is not certain about it, the test given on Sunday will indeed come as a surprise.

[6] It is important to note that K7, which holds in CFS and, therefore, also in (CFS + P*), ensures the knowledge of all the logical truths holding in the initial system (i.e. CFS). It does *not* ensure (or exclude) knowledge of the logical truth holding in the new system (CFS + P*). In particular, K7 does not imply K(P*).

5.4.2 Adding K(P*) to (CFS + P*)

Remark 5.4.2 CFS + P* + K(P*) is actually equivalent to CFS + K(P*)

As controversial as it seems, adding K(P*) ("the student does know that there is going to be a test next week") to CFS leads to a very interesting discussion.

In Section 5.1 we have showed that if K(P*) is true, (4) is also true, and therefore the bootstrapping reasoning thereby the student concludes the impossibility of a surprise-test on Sunday is sound.

However one should not forget that this is just the bootstrapping part of the student's entire reasoning. The paradox states the student is able to know not only that a surprise-test cannot be given on Sunday, but also that he is able to know that a surprise-test cannot occur on any day. That is, *a surprise-test cannot be given at all.*

We should study whether this latter conclusion is justified or not.

Once the student has concluded ~SSun, his reasoning goes on as follows:

"Let me suppose now that the test is given on Saturday. I have just concluded before that the test cannot be given on Sunday, and I keep knowing that on Friday. This means I would be able to know on Friday that the test must be given on Saturday. But then, no surprise-test can be given on Saturday either."[7]

This stage of the student's reasoning appears to take the following form:

$$\text{TSat} \rightarrow \sim\text{TBFri} \quad (1)$$

$$\sim\text{TBFri} \rightarrow \text{KFri}(\sim\text{TBFri}) \quad (2)$$

$$\text{TSat} \rightarrow \text{KFri}(\sim\text{TBFri}) \quad (3)$$

$$\text{KFri}(\sim\text{TBFri}) \rightarrow \text{KFri}(\text{TSat or TSun}) \quad (4)$$

---

[7] We must remark once again that this phrase is just a shortcut that hides a lot of implicit assumptions and deductions.

$$\text{KFri}(\sim\text{TSun}) \quad (5)$$

$$\text{TSat} \rightarrow \text{KFri}(\text{TSat}) \quad (6)$$

$$\text{KFri}(\text{TSat}) \rightarrow \sim\text{SSat} \quad (7)$$

$$\sim\text{SSat} \quad (8)$$

Again, the formal reasoning presented above is flawed. This time, the "guilty" clause is the clause referred here as (5).

This clause states that the student knows on Friday there is not going to be any test on Sunday. Although this *might* seem to result from the bootstrapping algorithm, it does not.

It is time for a reminder: the bootstrapping reasoning leads to the conclusion that a *surprise-test* will not be given on Sunday. This does not mean, however, that a ***non***-surprise-test cannot be given on Sunday.

The fact is that, using the given axioms, there is no way to conclude KFri(~TSun).

The student's reasoning is unsound not only in the CFS system, but also in the enhanced system CFS + K(P*).

But does the student's reasoning have any chance *to become* sound if we add some other axiom(s) to the system CFS + K(P*)?

Suppose we add K(~TSun) as axiom to the system CFS + K(P*).

In the new system CFS + K(P*) + K(~TSun), (5) becomes true and therefore the student can rightly conclude that a surprise-test cannot be given on Saturday. That is, K(~SSat) has become a provable theorem.

However the same kind of problem as before arises once again when the student moves to the new stage and tries to deduce K(~SFri), which is not provable in the system CFS+K(P*)+K(~TSun)

In order to conclude that, a new axiom should be added to the system CFS+K(P*)+K(~TSun): K(~TSat)

In the new system (CFS+K(P*)+K(~TSun)+ K(~TSat)), K(~SFri) is a theorem. But this not validates the entire reasoning of the student.

How many axioms do we have to add to the

system CFS + K(P*) in order to make the *entire* reasoning sound? We need no more and no less axioms than the number of days in the next week!

To be precise, we need to assert specifically for each day that we know there is not going to be any test on that day. But this amounts to asserting that we know there is not going to be any test next week, which is equivalent to asserting K(~P*)!

The formal system needed to acquire the reasoning soundness is equivalent to:

CFS + K(P*) + K(~P*)

But such a system is inconsistent, since the laws of knowledge asserted in CFS prevent someone from knowing both P and ~P*.

In an attempt to enhance the formal system such that the student's reasoning can become sound, we have ended with a contradictory system. This means the student cannot logically conclude that there is not going to be any surprise-test next week. At least, *not following the reasoning line suggested by the paradox*. But could there be any other way to enhance CFS such that K(P) becomes a valid conclusion?

The following theorem, *provable in CFS*, provides for a negative answer.

Theorem ~K(P)

In order to prove the theorem, we first prove the following lemma:

Lemma K(P) → [for any d, K(SBd)]

Suppose K(P) is true. We prove by induction K(SBd), with *d* standing for any day.

First, let *d* be Sun.

K(SBSun) is true, because P means exactly SBSun.

Suppose now *d* is an arbitrary day and K(SBd) is true. We must prove K(SByd) is also true.

The following implication holds in CFS:

K(SBd) → K(~SByd →Kyd(Sd)) (1)

(1), along with K7, implies:

K(SBd) → K(SByd or Kyd(Sd)) (2)

But it can easily be proved in CFS that ~Kyd(Sd), along with K7, implies:

K(~Kyd(Sd)) (3)

From (2) and (3) (along with K7), it can be concluded:

K(SBd) → K(SByd)

This way the lemma has been proven by induction.

We can prove now the theorem ~K(P).

Suppose K(P) was true and let *d* be an arbitrary day. Using the previous lemma for *yd*, we know:

K(SByd)

Hence we conclude K(~Sd)

But *d* was arbitrarily chosen. This means the following conjunction holds:

K(~SSun) and K(~SSat) and K(~SFri) and ...

Using the axiom K2:

K(~SSun and ~SSat and ~SFri) and ...), or:

But we assumed K(P) was true. So, we actually proved:

K(P) → K [Any d, ~Sd]

Using K1:

K(P) → K(Any d, ~Sd) → [Any d, ~Sd] (4)

But, on the other hand, K1 and the definition of P ensure:

K(P) → P → [Some d, Sd] (5)

From (4) and (5):

K(P) → [(Any d, ~Sd) and (Some d, Sd)]

K(P) → FALSE

Therefore, K(P) is false, so ~K(P) is true, q.e.d.

6. Conclusions

Accepting that ~K(P*) is actually all that is needed to solve – or, better said, to *dissolve* – the student's reasoning and, along with it, the whole paradox. But why seems that hard to admit such a simple explanation (as proved by the great amount of theories inspired by the paradox)?

In our opinion, the puzzlement raised by the surprise-test paradox stems from a confusion between *hearing someone uttering a statement S* and *knowing the truth of S*. Students (and, along with them, some paradox critics) *believe* that, since the professor says he is going to give a surprise-test next week, they can *know* at least one thing: some kind of a test (whether surprise-test or not) will certainly be given.

But no matter what he tells the student, the professor can in fact give a test next week or not.

If the professor does give a test next week, then there are two possibilities: the student knows there is going to be a test or he does not know that.

Assuming the student has some sort of crystal ball and he knows for sure there is going to be a test (but he does not know on what day precisely), he is right to conclude that if the test is given on Sunday, it is not going to be a surprise-test. Moreover, he could conclude that if the test is given on any other day, the test is a real surprise-test. Confronted with a student having *some* knowing about the future (i.e. K(P*)), the professor must schedule the test before Sunday if he is really serious about preparing a *surprise-test*.

If the student lives in a "normal" world, where no certain knowledge about the future is possible, he does not know for sure there is going to be a test until the test is actually given. Such a student is not logically entitled to conclude anything. Not even that a surprise-test may not be given on Sunday. The professor is free to choose whatever day for a surprise-test, *including Sunday*.

The most interesting result obtained in our article is probably the one provided by the theorem proved of Section 5.4.2:~K(P)

The theorem should be read like this: there is no possible way a student could know that the sentence uttered by the professor is true.

The professor's utterance is a magnificent and very intriguing example of a possibly true sentence that is not knowledgeable in any way.[8]

We can formulate our final conclusions using the paradigm of possible worlds:

In some of the possible worlds, P is false and K(~P) is true.

In other possible worlds, P is false and K(~P) is also false.

Still, in other possible worlds P is true, but K(P) is false.

However, there are no possible worlds in which P is true, and K(P) is also true.

REFERENCES

POUNDSTONE and WILLIAMS, **Labyrinths of Reason: Paradox, Puzzles and the Frailty of Knowledge**, PENGUIN BOOKS, 1991.

[8] P is, after all, very much alike to the following statement: "The truth of this very sentence cannot be known."