Formalizing the unexpected hanging paradox: a classic surprise

Polina Vinogradova[1][0000−0003−3271−3841]

IOG `polina.vinogradova@iohk.io`

Abstract. In this work, we define a novel approach to the formalization of the unexpected hanging paradox, sometimes called the surprise examination paradox, mechanized in the Coq Proof Assistant. This paradox requires the definition of the notion of a *surprise* event, which, for the purposes of this paradox, is usually interpreted as the inability to predict what day a specific event takes place. As in existing work, the use of constructive logic allows us to distinguish between possibility and certainty. We make the observation that an inevitable, but unexpected, event requires there being strictly more than one possible day on which it can occur, and define surprise accordingly. We formalize the paradox using this interpretation of surprise. Additionally, we specify a family of predicates for making judgements about whether a hanging occurs on a particular day, which are in accordance with the paradox constraints. We demonstrate that this family is inhabited by classical predicates. We then show that a classical version of our definitions lets us formally make the desired intuitive conclusions previously made in existing work using self-referential predicates. We assert that this offers an unexpected, but satisfying resolution to the paradox in accordance with our intuition. We compare our definition to an existing, weaker, interpretation of surprise, giving an analysis of how it interplays with the use of constructive logic.

Keywords: surprise examination · paradox · unexpected hanging · formalization · Coq · constructive logic.

1 Introduction

The unexpected hanging paradox, also known as the surprise examination paradox, is a logical paradox introduced in the the Mind philosophical journal in 1948 [14], and popularized by the Scientific American Mathematical Games column author Martin Gardner, discussed in his work [6]. It describes the notion of a future event that is both certain, and not possible to predict the exact day of occurrence of. It is formulated as follows:

> A judge tells a condemned prisoner that he will be hanged at noon on one weekday in the following week but that the execution will be a surprise to the prisoner. He will not know the day of the hanging until the executioner knocks on his cell door at noon that day.

Having reflected on his sentence, the prisoner draws the conclusion that he will escape from the hanging. His reasoning is in several parts. He begins by concluding that the "surprise hanging" cannot be on Friday, as if he has not been hanged by Thursday, there is only one day left − and so it won't be a surprise if he's hanged on Friday. Since the judge's sentence stipulated that the hanging would be a surprise to him, he concludes it cannot occur on Friday.

He then reasons that the surprise hanging cannot be on Thursday either, because Friday has already been eliminated and if he has not been hanged by Wednesday noon, the hanging must occur on Thursday, making a Thursday hanging not a surprise either. By similar reasoning, he concludes that the hanging can also not occur on Wednesday, Tuesday or Monday. Joyfully he retires to his cell confident that the hanging will not occur at all.

The next week, the executioner knocks on the prisoner's door at noon on Wednesday − which, despite all the above, was an utter surprise to him. Everything the judge said came true.

Existing formalization efforts attempt to address questions like "how can we formally define surprise within the conditions of this paradox?", "where is the flaw in the reasoning of the prisoner?" and "was it contradictory for the prisoner to have been hanged on Wednesday?". There is work on tackling these questions in multiple different branches of philosophy and mathematics. An extensive review of the existing approaches is given in [2].

The definition of surprise as the inability to deduce beforehand the date of the test was first introduced in [16]. A statistical approach to the problem of prediction in this context is discussed in [10]. A proposed solution in the field of epistemology with the use of modal logic, is presented in [7]. An approach involving Kripke semantics, employing the notion of persuasion, is in [9].

The use of constructive logic, e.g., appealing to Gödel's incompleteness, was first applied to the paradox in [5], then in [11], and most recently in [1] [15]. The first two rely on reasoning via the provability operator Pr to indicate a possibly non-classical statement, while the others assume the underlying logic to be itself constructive. This approach aligns most closely with ours.

The work [8] investigates the relation between the conclusions of the non-classical logic approaches and those from epistemology, as well as presenting four distinct approaches to formalizing the paradox, from which we take inspiration.

Here, we give formal and mechanized descriptions of related but distinct aspects of the paradox: the conditions of the paradox, and a family of predicates that can be used to make judgements about the hanging day in accordance with the paradox constraints. We note that this family of predicates may contain both classical and intuitionistical members. With the right definition of "knowledge of the hanging day", a classical inhabitant allows us to make consistent judgements, as well as judgements about inconsistencies, which are similar in content to those in the self-referential formalization presented in [11].

We take as our base assumption that the constraints of the paradox are fixed and correctly conveyed to the prisoner. This includes a fixed, a-priori selected execution date. Like the works we cited above, we do all our reasoning in constructive logic, leaving room for uncertainty, or possibility, by dropping the law of excluded middle. We additionally, make the assumption that the prisoner is able to continue drawing conclusions about the paradox on days after the hanging occurs. This is actually more in line the surprise examination version of this paradox, wherein the students are both surprised and alive after the exam happens.

In our formalization we do not make use of modal or temporal logic (which is the approach in [7]). Instead, we take advantage of the expressivity of the dependently typed logic of Coq to build our formalization, parametrized by weekdays in the appropriate way. Investigating the possibility of re-defining our dependently-typed functions as clauses in modal logic remains the subject of future work.

We chose to use a proof assistant (Coq, see [3]) to take a more high-assurance look at the interplay between the seemingly simple conditions of this conundrum. There is precedent for the use of proof assistants to tackle philosophical investigation, with the most striking and recent example being a refinement of Kant's categorical imperative [13].

The contributions of this paper are as follows:

(i)   A definition of surprise, together with the paradox constraints, without self-reference, Sections 4 6, reflecting the following natural language statement: "a hanging has not occurred on or before a given day, and there exist at least two distinct future days on which a hanging is possible". We also give an analysis of how this definition aligns with our intuition;

(ii) The construction of a three-parameter predicate returning a proposition about whether or not a hanging happens on *a given day*, given that we know whether or not a hanging happened on days up and including the parameter day *today*, and a day on which the hanging *actually happens*, see Section 5;

(iii) A proof that a classical instantiation of the predicate in (ii) is in accordance with our paradox specification, see Section 6;

(iv) A proof that a hanging having not occurred by Thursday does not let us conclude whether one will occur Friday or not, regardless of the specific definition of the predicate in (ii), Section 6;

(v)   An alternate, weaker formalization of surprise (Section 7). We give an analysis of how our definition compares to this one, and why it gives us insufficient reasoning power in the case that the surprise reasoning predicate is non-classical, but in the case that it is classical, allows us to reason ourselves out of the hanging entirely by concluding a sure-thing Friday hanging on Thursday;

(v)   An associated mechanization, in the Coq Proof Assistant, of the formal definitions and proofs in (i)-(v).

For our code, see https://github.com/polinavino/unexpected_hanging/blob/master/unexpected_hanging.v.

2   Mechanizing and the Paradox

Coq is a proof assistant that offers a dependently typed formal language. It is capable of verifying formal user-defined proofs of propositions, as well as supporting the automation of certain kinds of proofs. The choice of Coq, as opposed to another proof assistant such as Agda, was based largely on the authors' familiarity with the system, as any dependently typed proof verifier that supports constructive logic would serve just as well for the purposes of this mechanization.

To formalize the paradox, we need to reason about days of the week on which the paradox could happen, so we begin by constructing a type weekDay, the terms of which are week days:

```
Inductive weekDay : Type :=
  | monday : weekDay | ... | friday : weekDay.

Inductive weekAndBefore : Type :=
  | dayBefore : weekAndBefore | someWeekDay : weekDay →
      weekAndBefore.
```

We define the type weekAndBefore, which represents all the weekdays in the type above, plus the Sunday that comes before. The purpose of this type is to represent all the days on which one can consider the possibility of surprise, differentiating it from the subset of days on which the hanging can occur. We also define comparison function <, which computes whether a given td : weekAndBefore is before d : weekDay, following real-life weekday logic, e.g. Sunday is before Monday.

It is important to emphasize here that $=$, $\geq$, $<$ are all *decidable* comparison functions on days — purely as a consequence of considering weekdays as totally ordered entities, even in constructive logic. Any predicates formulated solely out of those comparisons together with logical connectives are also decidable, with the implication that provability, knowledge, and truth are all the same for such predicates, leaving no room for uncertainty. Therefore, solutions of the paradox constructed out of only such decidable predicates (e.g. [11]) are operating in classical logic.

For this reason we define a (for now) abstract function, which specifies a subset of days of the week on which a hanging occurs.

```
Variable hangingOnDay : weekDay → Prop.
```

We discuss, in Section 5, what properties and additional parameters of such a function ensure that it conforms to the constraints of the paradox. Next, we define a predicate that formalizes the notion that no hanging has occurred yet (up to and including the parameter day td, representing *today*). This says that for any day d, if it is before today td, no hanging happened on d:

```
Definition noHangingYet (td : weekAndBefore) :=
  ∀ d, td ≥ d → ¬ hangingOnDay d.
```

We use the double negation ¬¬ `hangingOnDay` d to formalize the statement that it is not possible to disprove that a hanging occurs on day d. That is, a hanging is *possible* on a given day.

3 Uniqueness of hanging.

Uniqueness of the hanging day plays an important role in the definition of surprise. We define the predicate `uniqueHanging` formalizing that "after a given day `td`, there can be at most one day on which a hanging occurs". So, `uniqueHanging dayBefore` states this about the entire week.

```
Definition uniqueHanging (td : weekAndBefore) :=
  ∀ d d', td < d ∧ td < d' →
  hangingOnDay d → hangingOnDay d' → d = d'.
```

The constraint, `uniqueHanging`, that a *provable* hanging day is unique is, in fact, equivalent to the constraint that a *possible* hanging day is unique. Negation of uniqueness is also implied by a stronger statement, `twoPossible`, which explicitly requires the presence of at least two possibilities for the hanging date. Note here also that this reasoning does not rely on any additional knowledge about the hanging predicate or the real day of the hanging.

```
Definition uniqueMaybe (td : weekAndBefore) :=
  ∀ d d',
  td < d ∧ td < d' →
  ¬ ¬ hangingOnDay d →
  ¬ ¬ hangingOnDay d' →
  d = d'.

Lemma uniqueMaybeEqv (td : weekAndBefore) :
  uniqueHanging td ↔ uniqueMaybe td.

Definition twoPossible (td : weekAndBefore) :=
  ∃ d d', td < d ∧ td < d' ∧ d ≠ d'
  ∧ ¬¬ hangingOnDay d ∧ ¬¬ hangingOnDay d'.

Lemma twoNotUnique : ∀ td,
  twoPossible td → ¬ uniqueHanging td.
```

The proof of this lemma relies on the decidability of d = d', together with modus tollens. This result seems wrong — we would expect a hanging to be unique — it's implicit in the description of the paradox. However, our definition of surprise, in fact, requires the non-uniqueness of a *future* hanging. If a hanging has already occurred in the *past*, we must still ensure that no additional hanging can happen in the future, making a past hanging unique.

4   A lack of surprise

Surprise is a hard concept to address, so we try to define, instead, what it means to be certain about when a hanging happens, given a collection of days on which it can happen. We define what it means for us to *know* that a hanging happened before today `td`:

```
Definition knowHanging (td : weekAndBefore) :=
  (∃ d, td ≥ d ∧ hangingOnDay d) ∧ (uniqueHanging dayBefore).
```

For this predicate to be true, the function `hangingOnDay` must be provably `True` for exactly one day `d` of the entire week, and `False` for all other weekdays, and this day `d` is before or on `td`. This predicate `knowHanging` should hold exactly when a hanging has already happened before today.

Now, let us consider the negation of these two conditions for days `d` after `td`:

```
Definition dontKnowHanging (td : weekAndBefore) :=
  ¬ ((∃ d, td < d ∧ hangingOnDay d)
    ∧ (uniqueHanging dayBefore)).
```

Representing that either there is no hanging, or it is not unique. Assuming no hanging has yet happened, this expresses surprise fairly well, however, it allows for the possibility that no hanging happens at all in the rest of the week, which should only be true if one had occurred before `td`. So, we must add to this definition a constraint that if we are able to prove that no hanging happens this week, we have proven `False` (we will separately handle the case for when a hanging did already happen).

Note here that while we want to say that on each future day `d`, proving a hanging occurrence should not be possible (i.e. ¬ `hangingOnDay d`), adding this to our definition will allow us to prove that no hanging happens at all. That is, the proposition that no hanging happens the entire week implies that no hanging is even possible, as

```
¬ hangingOnDay d → ¬ (¬ ¬ hangingOnDay d)
```

This is inconsistent with the intuition of the paradox.

We notice that both `uniqueHanging td` and its negation are trivially satisfied whenever ¬ (∃ d, td < d ∧ hangingOnDay d). For this reason we use the stronger `twoPossible td` in our definition of surprise, which contradicts the possibility of there not being a hanging at all, and guarantees two possible days.

So, regardless of when the hanging actually happens, we define what it means for surprise to being possible after `td` as

```
Definition surprise (td : weekAndBefore) :=
  (noHangingYet td) ∧ (twoPossible td).
```

5   The hanging predicate

According to the definition of the paradox, a Wednesday hanging satisfies the constraints. However, the spirit of the paradox seems to suggest there is nothing special about a Wednesday hanging. Here, we explain how to parametrize our predicates by different possible hanging days.

   We already parametrize the function `surprise` by today `td`, representing our inability to predict a hanging day (i.e. choose a unique hanging day) after `td`. The function `surprise` builds propositions about whether a hanging day happened on some day `d` by using the predicate `hangingOnDay`. We note that this predicate must actually be a different predicate depending on what day `td` is as well, since we must vary it to indicate that we always have full knowledge of all days before today, as well as full knowledge of the entire week whenever the actual hanging day has passed. So, all in all, we are interested in a function

```
hangingOnTodayIsReasoningAbout hang td d : Prop
```

that satisfies the paradox constraints. It has three parameters:

(i) `hang : weekDay` is the day on which the hanging actually occurs. Once today is or after this day, surprise should no longer be possible, but the paradox conditions may not be violated);
(ii) `td : weekAndBefore`, which is the day that is "today", i.e. the day on which the prisoner is making the judgement about the hanging day. For each day up to and including today the proposition about whether a hanging has occurred or not is classical;
(iii) `d : weekDay`, the day about which the prisoner is making the judgement of whether a hanging occurred on this day or not (e.g. tomorrow), but should not be able to prove or disprove the occurrence of a hanging on, unless this day is prior to `td`.

   Such a parametrized predicate replaces the previous `hangingOnDay` predicate in our definitions. To accommodate this change, we also parametrize all other predicates used in the definition, e.g. `noHangingYetparam hangingOn hang td`.

6   Paradox statement

We finally give the full statement of the constraints to be placed by the description of the paradox on the hanging predicate `hangingOnTodayIsReasoningAbout` discussed above.

```
Definition twoPossiblePRDXparam
    (hangingOn : weekDay → weekAndBefore → weekDay → Prop)
    (hang : weekDay) (td : weekAndBefore) :=
  (td ≥ hang ∧ (hangingOn hang td hang)
```

```
        ∧ uniqueHangingparam (hangingOn hang td) dayBefore)
    ∨
    (td < hang ∧ noHangingYetparam td
        ∧ twoPossibleparam (hangingOn hang td) td).
```

We want to define a `hangingOnTodayIsReasoningAbout` such that the following lemma holds true:

```
Lemma hangingFuncOk : ∀ hang td,
    ¬ (td = (someWeekDay thursday) ∧ hang = friday) →
    twoPossiblePRDXparam
        (hangingOnTodayIsReasoningAbout hang td) td.
```

We explicitly exclude being surprised on Thursday by a Friday hanging, as it appears to legitimately be a situation devoid of surprise (see discussion below). The statement of the paradox can be interpreted as follows:

> The executioner plans the hanging to be on the day `hang`. He tells the prisoner that in order to, on each day of the week `td`, draw the correct conclusion about whether the actual hanging day `hang` coincides with a particular week day `d`, any predicate `hangingOnTodayIsReasoningAbout` he uses to make this conclusion must satisfy the constraints of the paradox, as required by lemma `hangingFuncOk`.

We now define the predicate:

```
Definition hangingOnTodayIsReasoningAbout hang td d : Prop :=
    (td ≥ hang → hang = d) ∧ (td < hang ∧ td > d → False)
    ∧ (td < hang ∧ td < d → (¬ ¬ hf hang td d)).
```

which says that if today is after the day of the hanging, this is a provable predicate when the day being reasoned about is the same as the hanging day. If today `td` is prior to the actual hanging day, the day about which we are making a judgement cannot have a hanging. Finally, if the hanging day is not yet reached, a hanging occurrence cannot be disproved for any future day.

In this definition, we are able to substitute variable definitions of `hf`. We observe that the value `True` instead of `hf hang td d` is suitable here. In fact, we prove that it works by defining it that way and proving the above lemma `hangingFuncOk`. This is indeed the culmination of our formalization: we have specified the paradox in a way that aligns with our intuition with respect to what days a hanging is a surprise, but does not require self-referential or non-classical definitions, or indeed, a uniquely defined predicate for making judgements about future hangings.

However, perhaps claiming that a hanging is going to happen on all future days, and we can prove it, does not sound like a satisfying resolution. We argue that it may well be — since in order for us to "know when the hanging happens", we must prove that it happens on exactly one day. If we prove that it happens

on multiple days in the future, this does not constitute *knowledge* of when the one (un)-expected hanging will happen.

An advantage of our formalization is, in part, that it appears that both provability and possibility of multiple future hangings can be used to define surprise. We only specify a classical predicate `hf` as Coq does not support straightforward definition of non-terminating functions. We conjecture (and aspire to show as part of future work) that it is possible to define `hf` to be strictly weaker than `True` for the relevant inputs, satisfying the paradox in a strictly intuitionistic way. Such a definition would align more closely with the intuition that all future days should have a *possibility* of hanging, however, resulting in similar conclusions. We also hope to specify the weakest possible `hf` that satisfies the paradox.

According to the paradox, it is not possible that by Thursday, no hanging has happened, but there are still two distinct possible days on which it can happen in the future.

```
Lemma cantBeSurpFriday hangingOnTodayIsReasoningAbout :
  ∀ hang,
  twoPossiblePRDXparam hangingOnTodayIsReasoningAbout hang
    (someWeekDay thursday)
  → noHangingYetparam hangingOnTodayIsReasoningAbout hang
    (someWeekDay thursday)
  → False.
```

The proof (see the code) is simple, since there is only one day (Friday) left in the week, and no hangings are possible on past days. This can be interpreted in the following way:

> If no hanging occurred by Thursday, any conclusion we may draw about whether a hanging occurs on Friday is incorrect, i.e. not consistent with the conditions of the paradox.

So, the paradox constraints, as define here, do not let us make a consistent statement about whether a hanging happens in this circumstance or not. This is, however, also a conclusion we may arrive at intuitively. With all other combinations of parameter days, there is no inconsistency.

Inductive reasoning in an attempt to conclude that a Thursday hanging is predictable on Wednesday does not work because we do not have enough data on Wednesday to conclude whether a hanging will be Thursday or Friday. That is, to discount Friday as a possibility of surprise hanging, we *need to know that there was no hanging Thursday*. Thus, the prisoner's attempt at reasoning himself out of the hanging appears faulty.

7   At least one possible day

.

Surprise requires that a future hanging is possible — on more than zero of the remaining weekdays after today. There is precedent [8] for defining surprise in a way that allows a Friday hanging to be a surprise in a consistent way. We specify this interpretation of surprise, together with the other conditions of the paradox (i.e. that once a hanging occurs, there will be no future hangings):

```
Definition onePossiblePRDXparam
    (hangingOn : weekDay → weekAndBefore → weekDay → Prop)
    (hang : weekDay) (td : weekAndBefore) :=
  (td ≥ hang ∧ (hangingOn hang td hang)
      ∧ uniqueHangingparam (hangingOn hang td) hang dayBefore)
  ∨
  (td < hang ∧ noHangingYetparam hangingOn hang td ∧
      ∃ d, td < d ∧ ¬ ¬  (hangingOn hang td) d).
```

where the first disjunct is the same as before, and the second one corresponds to "today is not yet Friday, and there is a possible day on which a hanging may happen", which we refer to as the `onePossible` definition of surprise. This is a strictly weaker definition than `twoPossiblePRDXparam`, as the last disjunct requires only one possible day to exist, rather than two distinct ones. So, the same predicate `hangingOnTodayIsReasoningAbout` satisfies these constraints as well.

No inconsistency is introduced here, in fact, the hanging can still be a surprise even if it happens on a Friday! The intuition behind this is: if no hanging happened by Thursday, it is still only possible to prove ¬ ¬ `hangingOn hang td friday`, from which we are not necessarily able to deduce that `hangingOn hang td friday`. However, defining `hf hang td d = True`, for all future days, if no hanging has yet occurred, immediately results in being able to prove a unique (future) Friday hanging when today is Thursday, which is, in this case, in conflict with the intuitive notion of surprise. We do not get the inconsistency we "intuitively" expect here.

Let us consider the possible, i.e. ¬ ¬ case, and what happens if we impose an additional constraint stating that exactly one *possible* hanging day implies that it *provably happens* on a specific day. A similar kind of reasoning is also explored in [8], with a similar conclusion to the one we draw here. The following predicated states that (i) there is a possible hanging day, and that (ii) uniqueness of hanging day possibility implies certainty of hanging on that day:

```
Definition existsUniqueHappens :=
  (∃ d, ¬ ¬ hangingOnDay d)
  ∧
  (∀ d d', ¬ ¬ hangingOnDay d
    → ¬ ¬ hangingOnDay d' → d = d')
  → ∃ d, hangingOnDay d.
```

Now, the following statement expresses that `existsUniqueHappens` lets us conclude that `hangingOnDay` must then be classical (the proof is in the associated code):

```
Lemma euhImpClassical :
  (uniqueHanging dayBefore) →
  (∃ d, ¬ ¬  hangingOnDay d) →
  existsUniqueHappens →
  (∀ d, ¬  hangingOnDay d ∨ hangingOnDay d).
```

This is the crux of the reasoning the prisoner engages in (informally) to arrive at the judgement that if a hanging has not happened by Thursday, it must happen on Friday. Note here that the inductive reasoning the prisoner uses to conclude that the hanging cannot ever be a surprise is, in some sense, superfluous — we can, without induction, prove that "if we can conclude from existence plus uniqueness of a possible hanging day, that it is certain that on that day, *our judgement about hanging occurring on any day must necessarily be classical*". The proof, again, makes use of the fact that the equality comparison d = d′ is classical. This definition leaves us with the following conclusions about defining surprise as having at least one possible hanging day:

(i) a definition of surprise with a non-classical `hf` function may not be strong enough to either expect a Friday hanging on Thursday, or to arrive at an inconsistency on Thursday; and

(ii) if we *were* to be able to conclude `existsUniqueHappens`, we can always figure out the unique hanging day at any point in the week.

Both possibilities appear problematic: (i) does not allow us to make a conclusion that we would like make according to intuition, and (ii) removes the possibility of surprise on any day of the week.

Note here that there is a major difference between one- and two-possible versions of surprise with respect to the situation of today being Thursday, and no hanging has yet occurred. This difference is that assuming `hf friday (someWeekDay thursday) friday = True`, rather than a only possibility of a hanging on Friday, breaks the one-possible paradox case, allowing us to prove things that go against our intuition.

This is because the one-possible case, assuming the `True` version of `hf`, allows us to prove a unique Friday hanging on Thursday without contradiction, but the two-case does not allow this. However, using a weaker `hf` in the one-case, for which we can only prove a hanging possibility, does not allow us to conclude that the hanging will definitely happen on Friday even when it is the only possible day.

So, for the one-possible case, we must *rely* on additional assumptions of classical reasoning to conclude surprise is not possible on Thursday, which leads to the conclusion that the prisoner arrived at. In the two-possible case, we can use the paradox definition itself, together with any suitable definition of the hanging predicate (classical or otherwise) to conclude that a Friday hanging will not be a surprise on Thursday.

8   Conclusion and Future Work

We have presented a formalization of the constraints of the unexpected hanging paradox parametrized by the day the hanging is planned, together with a family of predicates (a distinct one for each day the hanging will actually take place) that lets us judge, according to the information the prisoner has, if a hanging is happening or not on a given past and future day from the perspective of each of the days of the week. This formalization appears to capture the conditions of the paradox and align with our intuition, without the need for self-reference. We mechanized this definition, together with some related proofs, in the Coq proof assistant, relying on the particularities of constructive logic to represent the uncertainty, or possibility. Crucially, however, we have demonstrated that even a classical instantiation of a hanging predicate can satisfy the paradox constraints (and relevant conclusions about them), if in a slightly odd way. A few key ideas were needed to achieve this.

First, we observe that knowing when a hanging happens requires that we are able to reason that a hanging happens on some day of the week, and that this day is unique. We define a lack of knowledge of when a unique hanging happens (i.e. the possibility of surprise) as the negation of this, together with the assumption that there exists day on which a hanging is possible. This approach to defining surprise does not allow us to conclude, on Thursday, whether or not a hanging will occur on Friday, but allows us to pass judgements about all other combinations of hanging day, today, and day about which we reason. We notice that our definition of hanging does not require us to adhere to non-classical reasoning if all that is needed to claim uncertainty about the hanging day is the ability to prove that it may happen on multiple different days, rather than one unique one.

We compare our definition to a formalization of paradox constraints with a slightly weaker definition of surprise. This definition requires at least one day to exist in the future on which we cannot disprove the hanging. We notice that a classical definition of the hanging predicate leads to be able to demonstrate a unique future hanging day before the week even starts, which is counterintuitive. However, a non-classical definition of the hanging predicate does not allow us to definitively conclude a Friday hanging on Thursday — only the possibility of it. We conclude that this is not a sufficiently precise definition.

As part of future work, we conjecture this paradox formalization could be further analyzed by way of considering its relationship to the axiom of choice. This is due to its (at least surface level) resemblance to the way the AC makes a connection between classical logic and a choice function [4] as well as arbitrary elements [12].

Another possible direction of future work that we have considered is looking into the possible applications of our definition of a future event whose timing is unpredictable, but guaranteed to be within a certain time frame.

References

1. Ardeshir, M., Ramezanian, R.: A solution to the surprise exam paradox in constructive mathematics. The Review of Symbolic Logic **5**, 1–8 (12 2012). https://doi.org/10.1017/S1755020312000160
2. Chow, T.Y.: The surprise examination or unexpected hanging paradox (1999). https://doi.org/10.48550/ARXIV.MATH/9903160, https://arxiv.org/abs/math/9903160
3. CNRS, contributors: Coq reference manual (2021), https://coq.inria.fr/distrib/current/refman/
4. Diaconescu, R.: Axiom of choice and complementation. Proceedings of the American Mathematical Society **51**(1), 176–178 (1975), http://www.jstor.org/stable/2039868
5. Fitch, F.B.: A goedelized formulation of the prediction paradox. American Philosophical Quarterly **1**(2), 161–164 (1964), http://www.jstor.org/stable/20009132
6. Gardner, M.: Unexpected Hanging Paradox and Other Mathematical Diversions. University of Chicago Press (1991)
7. Halcrow, W., Holliday, W.: Simplifying the surprise exam (2016)
8. Halpern, J.Y., Moses, Y.: Taken by surprise: The paradox of the surprise test revisited. Journal of Philosophical Logic **15**(3), 281–304 (1986), http://www.jstor.org/stable/30226356
9. Harrison, C.: The unanticipated examination in view of kripke's semantics for modal logic (1969)
10. Kim, B., Vasudevan, A.: How to expect a surprising exam. Synthese **194**, 3101–3133 (2017)
11. Kritchman, S., Raz, R.: The surprise examination paradox and the second incompleteness theorem (2010). https://doi.org/10.48550/ARXIV.1011.4974, https://arxiv.org/abs/1011.4974
12. van Lambalgen, M.: Independence, randomness and the axiom of choice. The Journal of Symbolic Logic **57**(4), 1274–1304 (1992), http://www.jstor.org/stable/2275368
13. Lindner, F., Bentzen, M.M.: A formalization of Kant's second formulation of the categorical imperative (2018). https://doi.org/10.48550/ARXIV.1801.03160, https://arxiv.org/abs/1801.03160
14. O'CONNOR, D.J.: PRAGMATIC PARADOXES. Mind **LVII**(227), 358–359 (07 1948). https://doi.org/10.1093/mind/LVII.227.358, https://doi.org/10.1093/mind/LVII.227.358
15. Ramezanian, R.: A constructive epistemic logic with public announcement (non-predetermined possibilities). CoRR **abs/1302.0975** (2013), http://arxiv.org/abs/1302.0975
16. Shaw, R.: The paradox of the unexpected examination. Mind **67**(267), 382–384 (1958). https://doi.org/10.1093/mind/lxvii.267.382