Formalizing the Unexpected Hanging Paradox: a Classical 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. Our 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, then specify a family of propositions representing beliefs about whether a hanging occurs on a particular day. We define what members of this family are in accordance with the paradox constraints, and demonstrate that this family is inhabited by classical propositions. We assert that this offers an unexpected, but satisfying resolution to the paradox, which agrees with our intuition, all without the need for self-referential predicates used in existing work. We compare our definition to an existing, weaker, interpretation of surprise, giving an analysis of how it interplays with the use of constructive logic, and allows the prisoner to reach an apparently faulty conclusion.

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 [15], and popularized by the Scientific American Mathematical Games column author Martin Gardner, discussed in his work [7]. It describes the notion of a future event that is both certain, and for which it is not possible to predict the exact day of occurrence. 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 in accordance with 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 [3].

The definition of surprise as the inability to deduce beforehand the day of the hanging was first introduced in [17]. A statistical approach to the problem of prediction in this context is discussed in [11]. A proposed solution in the field of epistemology with the use of modal logic, is presented in [8]. An approach involving Kripke semantics, employing the notion of persuasion, is in [10].

The use of constructive logic, e.g., appealing to Gödel's incompleteness, was first applied to the paradox in [6], then in [12], and most recently in [1] and [16]. The first two rely on reasoning via the provability operator Pr to indicate a possibly non-decidable proposition, while the others assume the underlying logic to be itself constructive. The latter approach aligns most closely with ours, as our formalization is also constructive, leaving room for uncertainty and possibility.

In [9], four distinct approaches to formalizing the paradox, from which we take inspiration, are presented. This work additionally discusses the relation between constructive logic approaches to resolving the paradox and those from epistemology, drawing parallels between the conclusions of the two.

Here, we give formal and mechanized descriptions of the following related but distinct aspects of the paradox: (i) a family of propositions representing beliefs about whether a given day is, or could be, the hanging day (ii) a family of propositions specifying how the constraints of the paradox are imposed on the beliefs in (i). Both collections are parametrized by the planned hanging day. We demonstrate that while formalization of the family of beliefs is done constructively, it is, in fact, inhabited by decidable members, specifying beliefs that are consistent with the constraints of the paradox. We argue that with the definition of "knowledge of the hanging day" we present, a classical inhabitant allows us to form consistent and intuitive beliefs about the hanging day. Moreover, these beliefs are similar in content to those in the self-referential formalization presented in [12], as well as to those represented by other inhabitants of the family that are consistent with the paradox constraints.

We chose to use the proof assistant Coq (see [4]) 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. Some of the most striking recent examples include a refinement of Kant's categorical imperative [14], as well as a formalization of Gödel's ontological argument [2].

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 day, which is not known to the prisoner. This day nevertheless affects the prisoner's beliefs about the hanging day on the planned day and thereafter, since the prisoner necessarily finds out the hanging does happen when the planned day comes. Our formalization specifies the prisoner's beliefs about the hanging day even on days after the hanging has occurred, as if the beliefs are actually of the onlookers, who, like the prisoner himself, are unaware of the planned hanging day ahead of time. We elaborate on this choice in the discussion.

Another feature of our formalization is that we do not make use of modal or temporal logic, which is the approach in [8]. Instead, we take advantage of the expressive Coq type system to parametrize beliefs by relevant data about the situation in which the beliefs are being evaluated, such as what day today is, and whether the hanging has already happened. This absolves us of the need to specify "future beliefs", since beliefs held on different days about whether the hanging occurs on a particular day are specified independently, forming a parametrized family of propositions.

The contributions of this paper are as follows:

(i) A definition of surprise, together with the paradox constraints, formulated without self-reference (see Sections 4 6), reflecting the following natural language statement: "if a hanging has not yet occurred on or before a given day, 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) A family of functions which return, for a chosen planned hanging day, a proposition representing a belief about whether or not a hanging *happens on a given day*, given that we know whether or not a hanging happened on days up to and including the parameter day *today*, see Section 5;

(iii) A proof that any member function of the family in (ii), satisfying a certain property, must also satisfy the constraints of the paradox, other than in the case that today is Thursday, and no hanging has yet occurred, Section 6;

(iv) A proof that a particular decidable inhabitant of the class in (ii) satisfies the paradox constraints, see Section 6;

(v) An alternate, weaker formalization of surprise (Section 7), reflecting one of the possible definitions discussed in [9], alongside an analysis of how it relates to the prisoner's reasoning and constructive logic;

(vi) 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 the Paradox

Coq is a proof assistant based on the typed programming language Calculus of Constructions, which also forms a constructive foundation for mathematics. Coq 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 hanging could happen, so we begin by constructing the type `weekDay`, the terms of which represent days of the week:

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

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

We also 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 a future surprise hanging, differentiating it from the subset of days on which the hanging can occur. We also define the 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 propositions formulated using solely those comparison operators together with logical connectives are also decidable, with the implication that provability, knowledge, and truth are all the same for such propositions, leaving no room for uncertainty. Therefore, solutions of the paradox constructed out of only such decidable propositions (e.g. [12]) are operating in classical logic.

To avoid defaulting to classical logic, 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 allow us to specify when exactly it conforms to the constraints of the paradox. Next, we define a function that outputs the proposition that no hanging has occurred yet, up to and including its parameter `td` representing *today*. In our formalization, a specific *today* represents the day on which beliefs about the hanging day are being formulated by the prisoner. The following parametrized proposition 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.
```

Having a valid proof of the negation of `hangingOnDay d` represents the natural language statement "the hanging cannot occur on day `d`", or more specifically, that it cannot occur without introducing inconsistency into our system. We can interpret this as "the occurrence of the hanging on day `d` is disproved".

We use the double negation `¬¬ hangingOnDay d` to formalize the statement that disproving that a hanging occurs on day `d` implies `False`. That is, occurrence of a hanging on that day cannot be disproved, and therefore is *possible* on the given day. Note here that a triple negation is equivalent to a single negation, as, in constructive logic,

```
(¬ hangingOnDay d) ⇔ (¬¬¬ hangingOnDay d)
```

Therefore, no hanging is possible on `d` if and only if no hanging occurs on `d`.

3 Uniqueness of the Hanging Day

Reasoning about the uniqueness of a hanging day plays an important role in the definition of surprise. We define `uniqueHanging`, which formalizes that "after a given day `td`, there can be at most one day on which a hanging occurs". The proposition `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 proposition `uniqueHanging td`, stating that a *provable* hanging day is unique is, in fact, equivalent to stating that a *possible* hanging day is unique. Non-uniqueness of a hanging day is also implied by a stronger statement, `twoPossible`, which explicitly requires the presence of at least two possible hanging days. Note here also that this reasoning does not rely on any additional information about beliefs about the hanging day, or the planned 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 these lemmas relies on the decidability of `d = d'`, together with modus tollens. This result seems wrong — it appears to say that believing there to be more than one possibility for a hanging day is the same as believing the hanging will indeed occur on more than one day — but uniqueness of the hanging day is implicit in the description of the paradox! Note, however, that our definition of surprise requires the non-uniqueness of a possible *future* hanging day. The nuance here is that if a hanging has already occurred in the *past*, we must define the paradox constraints in a way that ensures that no *additional* hangings can happen in the future, making a past hanging remain unique.

4 A Lack of Surprise

Surprise is a hard concept to make precise, so we 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 to be true for a given `td`, the proposition `hangingOnDay td` must be provable for exactly one day `d` of the entire week, and `False` for all other weekdays, and this day `d` is on or before `td`. The proposition `knowHanging td` is provable whenever a hanging has already happened before today.

Now, let us consider the negation of these two conditions for days `d` after `td`, representing that either there is no hanging, or it is not unique:

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

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`. That is, `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.

Note here that one might be tempted to define surprise as the notion that on each future day `d`, proving a hanging occurrence should not be possible (i.e. `¬ hangingOnDay d`). As we showed earlier, this ensures that not only is the future occurrence of a hanging disprovable, but so is any possibility of a future hanging. Moreover, defining a proposition that ensures a hanging is not possible on all future days will allow us to prove that no hanging ever happens. This is contrary to the judge's announcement.

Regardless of when the hanging actually happens, we can define what it means for surprise to be possible after `td` as:

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

As this says nothing about when a hanging does actually happen, we must now introduce the planned hanging day into our reasoning. 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. Next, we explain how to accommodate this by modifying the hanging function with additional arguments.

5 The Hanging Function

We used the unspecified function `hangingOnDay` in our earlier definitions to simplify some preliminary reasoning about the nature of uniqueness, possibility, and surprise in this paradox. We now give the type of a modified hanging function, parametrized in a way that will allow us to formalize the conditions under which such a function defines a family of beliefs that are consistent with the paradox:

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

It has four parameters:

(i) `hf : weekDay → weekAndBefore → weekDay → Prop` is a function that constructs beliefs about all future hanging days, if no hanging has yet occurred;

(ii) `hang : weekDay` is the day on which the hanging *actually occurs*, as planned by the executioners. Once today is on or after this day, surprise should no longer be possible, but the paradox conditions may not be violated;

(iii) `td : weekAndBefore`, which is the day that is "today", i.e. the day *on which* the prisoner is forming a belief about the hanging day;
(iv) `d : weekDay`, the day *about which* the prisoner is forming the belief regarding whether a hanging occurred on this day or not (e.g. tomorrow).

This function replaces the `hangingOnDay` function in our definitions. To accommodate this substitution, we also parametrize all other functions used in the definition, e.g. `noHangingYetparam hangingOn hang td`. The function is defined as follows:

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

which says that if today is after the day of the hanging, and the day being reasoned about is the same as the hanging day, this is a provable proposition. If today `td` is prior to the actual hanging day, the day `d` cannot have a hanging on it when this day is before today. Finally, it says that for any today, if `¬ hf hang td d`, then either the hanging already happened, or `d` is in the past.

The parameter `hf` is itself a parametrized function used to represent beliefs about the hanging day. The point of this parameter is to demonstrate that *multiple definitions* of beliefs about a future hanging can actually be admissible as satisfying paradox constraints. To characterize when such functions are admissible, we give the complete definition of the paradox.

6 Paradox Statement

The function `twoPossiblePRDXparam`, below, *assesses the beliefs* of the prisoner, passed via the parameter `hangingOn`, to see if they are in accordance with the announcement of the judge that there is a surprise hanging this week, given that the hanging actually takes place on the pre-planned day `hang`:

```
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).
```

Given a planned hanging day `hang`, and a today `td`, the proposition constructed by this function says that either:

(i) if the planned hanging day is in the past, it must be unique across the entire week, or

(ii) if the planned hanging is in the future, there are at least two future days on which the hanging is possible.

Now, the following lemma:

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

formalizes the statement that `hangingOnTodayIsReasoningAbout` is a hanging function that specifies the prisoner's beliefs *in accordance with the paradox constraints* for any planned hanging day, on any today, given that the `hf` is *any* function satisfying a particular constraint. This constraint states that "given that today is `td`, a hanging cannot occur on the day `d` implies that either the planned hanging day is in the past, or the day `d` is itself in the past.

We additionally exclude being surprised on Thursday by a Friday hanging, as it is both formally and intuitively a situation devoid of surprise. Any conclusion we are able to draw on Thursday about whether a hanging occurs on Friday is incorrect whenever no hanging has occurred yet. For any hanging function satisfying the paradox constraints, this is expressed in the following lemma (see the code for the proof):

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

The `hangingFuncOk` lemma states that a specific family of hanging functions represents beliefs that are in accordance with the paradox, each parametrized by some `hf`. In general, `hf hang td d` need not be decidable. In fact, we can equivalently re-state the constraint as "if a hanging has not already happened, then in must be *possible* on any future day `d`,

```
    (td < hang ∧ td < d) → ¬¬ (¬ hf hang td d)
```

where possibility is expressed via double negation. The paradox intuitively suggests the importance of the distinction between *possibility* and *provability* of a future hanging, which is made in our work through the use of constructive logic. However, to reap the benefits of the formalization we propose, this is not required. We leave it as future work to construct an example of a function with which to instantiate `hf` to highlight the distinction, as Coq does not support straightforward definition of recursive functions that are not guaranteed to terminate.

The crux of our paradox analysis is that the hanging function can indeed be instantiated with a decidable function, e.g.,

```
   hf hang td d := True.
```

This follows immediately from the fact that arbitrary propositions can be proven from the premise `False`. With this instantiation, all propositions returned by the hanging function, as well as the paradox formalization itself, are decidable (recall here that weekday comparisons are always decidable). Consequently, we can prove that the hanging actually happens multiple times in the future, unless it has either already happened in the past, or it hasn't, and today is Thursday.

This is counterintuitive — however, recall that the definition of "*knowing* when a hanging happens" requires that there is a *unique* day for which we can prove that the hanging happens. We can only prove that there is a unique hanging day (or equivalently, a unique possible hanging day) when it's either in the past, or via contradiction (when Friday is the only remaining option). Recall here that we showed earlier that uniqueness of possible and provable hanging days is equivalent.

The inductive logic the prisoner uses to reason his way out of the hanging does not apply within our formalization. The argument that a hanging will necessarily happen on Friday requires the precondition that it has not happened by Thursday. For all other days, we are only ever able to prove that no hanging happened on a given day if that day is in the past (or a hanging already happened). So, if today is Wednesday or earlier in the week, and no hanging has occurred, we are not able to prove that no hanging happens Thursday. Therefore, earlier in the week, we do not have sufficient information to conclude anything about a Friday hanging that would violate the paradox.

Thus, the prisoner's attempt at reasoning himself out of the hanging appears faulty — which aligns with the premise of the paradox that a hanging does indeed occur. However, we argue that this conclusion is possible with the following weaker definition of surprise.

7 At Least One Possible Day

In any formalization of the paradox, there appear to only be the following reasonable options for what conclusion can be made about a Friday hanging on Thursday: it is either (1) provable, (2) not disprovable, i.e. possible, or (3) disprovable. We have explored a formalization where it is disprovable, and for this reason, excluded from the domain of definition of beliefs consistend with the paradox. Here, we will look at (1) and (2), which are not mutually exclusive, but an interesting distinction nonetheless.

Surprise requires that a future hanging is possible — on more than zero of the remaining weekdays after today. There is precedent [9] 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 in the following way:

```
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 in the two-possible definition, and the second one corresponds to "if the hanging has not yet happened, there is a possible day on which a hanging may happen in the future", which we refer to as the *one-possible* 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 family of hanging functions as for the two-possible version 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! This puts this formalization in category (2), a possible Friday hanging. 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, we also cannot explicitly restrict making this deduction, i.e. introduce:

```
  ¬ (¬¬ hangingOn hang td friday → hangingOn hang td friday)
```

as we can then immediately prove `False`. Tautologically, it does not make sense to have the possibility of something when it is definitely not happening. So, we do not introduce such a constraint.

Let us consider what happens if we impose an additional constraint stating that having *exactly one possible hanging day* implies that it *provably happens* on that specific day. This changes this formalization from category (2) to (1) above, as we can now prove the Friday hanging. The following defines a proposition stating 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 lemma expresses that `existsUniqueHappens` lets us conclude that `hangingOnDay` must then be decidable (the proof is in the associated code):

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

Concluding provability from possibility within the confines of the paradox definition is the crux of the reasoning the prisoner engages in (informally) to arrive at the belief that if a hanging has not happened by Thursday, it must happen on Friday. If we adhere to the definition of knowledge we presented earlier, this violates the intuition of what surprise should mean — i.e. the inability to pick a unique and provable hanging day in the future, by making it possible to predict a Friday hanging on Thursday.

If we admit this definition of surprise, the prisoner then has grounds to conclude that the promise of surprise across the entire week is a hoax, and reason himself out of being hanged. Note that no inductive reasoning is actually needed here. There is a future day of the week for which a prediction can be made, and this already contradicts "hanging will be a surprise", allowing us to prove anything from this contradiction. We can make the following conclusions from the one-possible surprise definition with and without the extra decidability premise:

(i) a definition of surprise using a constructive function (as in category (2) above) 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`, and reason using decidable propositions (as in category (1) above), the paradox constraints would allow us to predict a Friday hanging on Thursday, such that it is not a surprise, with no contradiction.

Both possibilities appear problematic: (i) does not allow us to make a conclusion that we would like to make according to our intuition, and (ii) gives a definition of surprise which allows us to construct a future hanging prediction.

The two-possible definition avoids these issues by having a stricter definition of the paradox that yields an inconsistency, rather than a prediction, for a Friday hanging. It does not rely on the decidability (or undecidability) of the hanging function to draw different conclusions. However, predictably, no paradox formalization should accommodate a Friday hanging prediction on Thursday, and thus this situation must be explicitly excluded from the paradox to draw any meaningful conclusions.

8 Discussion and Future Work

The goal of this work was to resolve the unexpected hanging paradox. We did so by mechanizing our formalization using the (constructive) calculus of inductive constructions, the underlying formal language of the proof assistant Coq. A few key ideas were needed to achieve this that do not appear to have previously been made explicit in existing literature.

We began by making a formal distinction between "a hanging is possible" and "a hanging happens" on a given day via the use of double negation. We then showed that asserting the uniqueness of a possible day hanging is equivalent to asserting the uniqueness of a provable hanging day.

Next, we formalized the concept of *knowing* the day a (unique) hanging event will occur when it is guaranteed to happen within a certain set of days, e.g. a particular week. We went on to formalize *surprise* as the negation of knowing a future hanging day, which led us to conclude that a *future* surprise hanging day (possible or provable) is necessarily not unique.

In order to define the paradox constraints, we first defined a family of "hanging functions", each of which specifies the beliefs a prisoner has on each weekday about the occurrence of a hanging, e.g. if the hanging has already occurred, it is unique across the entire week. Each function in this family corresponds to the particular parameters of a planned hanging day, and a function representing beliefs about a future hanging day if the planned hanging has not yet occurred.

The paradox constraints themselves are parametrized by the planned hanging day, as well as what day today is. They constitute a proposition that asserts that the hanging function beliefs are in accordance with the the announcement of the judge. We specified the family of hanging functions which ensure the paradox constraints are satisfied, for any planned hanging day (with the exception being that a Friday hanging can never be a surprise on Thursday). These conclusions aligned with our intuition. Somewhat surprisingly, however, we were able to show that a decidable instantiation of the hanging function satisfies the constraints as well as our intuition — without the need for constructive logic or self-reference. The reason for this is our definition of knowledge as the ability to select a unique provable hanging day.

We went on to contrast this satisfactory surprise formalization with a weaker one. In this version, depending on whether classical or constructive logic was used, the prisoner either had the opportunity to reason his way out of the hanging by disproving surprise, or was unable to conclude that a Friday hanging is provable on Thursday when no such contradiction loophole arose. We argue that the options of possible, provable, and disprovable hangings on a Friday given that today is Thursday define three categories of approaches to formalizing this paradox, and we have explored each here.

The final point we want to address about this formalization is that beliefs about the hanging can be specified for weekdays after the hanging. This aspect of our approach is actually more in line the surprise examination version of this paradox, wherein the students are both surprised and alive the rest of the week after the exam happens, and continue to have beliefs about the examination day. The effect of choosing one approach over another on the interpretation of the paradox is not significant. It amounts to constraining the parameters of both the paradox constraints and the hanging function to the "todays" that precede the planned hanging. All the noteworthy reasoning we do is from the perspective of "todays" on which the hanging has not yet happened. However, we chose to allow reasoning after the hanging for a cleaner and more complete formalization.

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 [5] as well as arbitrary elements [13].

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 discrete and unpredictable, but guaranteed to be within a certain time frame.

Acknowledgements I would like to thank my awesome graduate school supervisors, Dr. Amy Felty and Dr. Philip Scott, as well as numberous colleagues at IOG, for listening to me ramble on about this paradox. I would especially like to thank Dr. Pieter Hofstra, may he rest in peace, for making the bold move of asking for a resolution of this paradox as a (surprise) bonus question on a computability theory exam. I could not stop thinking about it ever since, until, hopefully, now.

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. Benzmüller, C., Woltzenlogel Paleo, B.: Interacting with Modal Logics in the Coq Proof Assistant. vol. 9139 (07 2015). https://doi.org/10.1007/978-3-319-20297-6_25
3. 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
4. CNRS, contributors: Coq reference manual (2021), https://coq.inria.fr/distrib/current/refman/
5. Diaconescu, R.: Axiom of choice and complementation. Proceedings of the American Mathematical Society **51**(1), 176–178 (1975), http://www.jstor.org/stable/2039868
6. Fitch, F.B.: A goedelized formulation of the prediction paradox. American Philosophical Quarterly **1**(2), 161–164 (1964), http://www.jstor.org/stable/20009132
7. Gardner, M.: Unexpected Hanging Paradox and Other Mathematical Diversions. University of Chicago Press (1991)
8. Halcrow, W., Holliday, W.: Simplifying the surprise exam (2016)
9. 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
10. Harrison, C.: The unanticipated examination in view of kripke's semantics for modal logic (1969)

11. Kim, B., Vasudevan, A.: How to expect a surprising exam. Synthese **194**, 3101–3133 (2017)
12. 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
13. 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
14. 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
15. 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
16. Ramezanian, R.: A constructive epistemic logic with public announcement (non-predetermined possibilities). CoRR **abs/1302.0975** (2013), http://arxiv.org/abs/1302.0975
17. Shaw, R.: The paradox of the unexpected examination. Mind **67**(267), 382–384 (1958). https://doi.org/10.1093/mind/lxvii.267.382