Constructive Logic Approach to Solving the Unexpected Hanging Paradox

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

Abstract—In this work we define a novel approach to formally specifying the unexpected hanging paradox, sometimes called the surprise examination paradox, using a constructive logical framework. We build this formal specification using the Coq proof assistant. This paradox requires the formalization 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 represent knowledge as provability. However, unlike many previous formalizations, we define a system, parametrized by days of the week, where intuitive conclusions can be justified formally and without inconsistency. We are able to achieve this by making a nuanced observation about the formal statement regarding the uniqueness of the occurence of the event, as well as the informal meaning of surprise. We believe that this offers a satisfying resolution to the paradox. We also Coq-formalize an existing unsatisfactory interpretation of surprise and compare it to ours.

Keywords—surprise examination, paradox, unexpected hanging, Coq, constructive logic.

I. Introduction

The unexpected hanging paradox, also known as the surprise examination paradox, is a logical paradox introduced in the Mathematical Games column of a 1963 issue of Scientific American [**?**]. It describes the notion of a future event that is both certain, and not possible to predict the exact day of occurence of, 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" can't be on Friday, as if he hasn't 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 hasn't 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?", "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 mathmatics, most notably a logical approach and an epistemological one. It appears, however, that the majority of resolutions of this paradox reach a conclusion espousing the impossibility of defining a consistent system with a coherent definition of surprise that is not self contradictory.

In the logical category, the approach most closely aligned with ours, where non-classical logic is used to tackle the problem, is described in []. There are a number of attempts preceding this one as well, [] []. An epistemological approach [], while from a different area of philosophy, is able to support reasoning that follows a similar structure to the logical reasoning, which are compared in [].

In the work on this paradox which precedes ours [] [], the use of constructive logic is usually done via having a proof operator Pr, which explicitly specifies that a proposition to which it is applied is a constructive statement. In our work, however, we assume the underlying logic to be constructive, and instead give formal proofs that certain propositions are classical.

Another distinction between existing work and our formalization is that we do not use modal or temporal logic (which is the approach in [**?**]). Instead, we take advantage of the expressivity of the dependent typed logic of Coq directly to achieve the formalization of surprise at different points in time, by parametrizing the definition of the surprise proposition by the day about which the proposition of surprise is constructed.

Because constructive logical is notoriously slippery, we chose to use a proof assistant to take a closer, more high-assurance look at the interplay between the seemingly simple conditions of this conundrum. We also give an analysis of the relation of the informal meaning of surprise to its formalization.

The contributions of this paper are as follows :

(i) a formal specification (in Coq) of the types and conditions of the that are shared among existing interpretations, using dependent types and constructive logic, Section III ;

(ii) a definition (in Coq) of surprise from a previous work, with an analysis of it in Section IV ;
(iii) a definition (in Coq) of surprise that we propose, Section V;
(ii) a formally verified proof that a Friday hanging is not a surprise, Section V;
(iii) a formally verified proof that our definition of surprise for this paradox is equivalent to the negation of the constraint that the hanging happens on exactly one specific day, Section VI;
(iv) an analysis of the unexpected hanging paradox offering a resolution without contradiction, Sections V VI VII VIII;

For our code, see [**?**].

Our definition of surprise on a day td is formulated to reflect the following natural language statement : a hanging has not occured on or before the day td, and there exist at least two distinct future days on which a hanging is possible (ie. it is not possible to disprove that a hanging occurs on either of those days). We argue that this accurately represents a lack of certainty about the exact day of the hanging, but a certainty in its inevitability, and does not contradict the other constraints of the paradox.

The crux of our paradox formalization analysis hinges on the observation that the uniqueness constraint is, in fact, never relevant to the formalization of the paradox — neither in reasoning about any day with a possibility of future hanging (ie. on which a hanging has not yet happened), nor about a day on which it has already happened. We go on to argue that the constraint that a hanging necessarily happens is also contradictory to our definition of surprise. Moreover, the looser constraint from previous work, saying that we *should not be able to disprove that a hanging happens on one of the days* is insufficiently strong, and is implied by our definition. However, the intuitive conclusion normally draw from it (that if a hanging is only possible Friday, it must happen then) is too strong.

II. Coq 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 has support for 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 would serve just as well for the purposes of this formalization.

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
  | tuesday : weekDay
  | wednesday : weekDay
  | thursday : weekDay
  | friday : weekDay.
```

We also define a 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 the two comparison functions,

```
isBefore, isOnOrAfter (td : weekAndBefore)
    (d : weekDay) : Prop := ...
```

which compute whether a given `td` is before (is on or after, respectively) `d`, following real-life weekday logic, eg. Sunday is before Monday. Both of these are classical comparisons, which we prove in our code. From here on, we use the notation $<$ for `isBefore`, and $\geq$ for `isOnOrAfter`.

We do not know what day the hanging happens on, but we can specify the type of a function that, given a day of the week, returns True if we can prove the hanging happened, and False if we can prove it did not. We reason about this function in the presence of preconditions that are formal interpretations of those described in the paradox. We leave this function as a variable :

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

We discuss the existence of such a predicate that works with in definition of surprise as part of future work. Next, we define a predicate that formalizes the notion that no hanging has occured yet (up to and including the parameter day `td`, representing *today*) :

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

This says that for any day `d`, if it is before today `td`, no hanging happened on `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.

III. Surprise parametrization and universal conditions

Notice that the `noHangingYet` function we introduced is actually a dependent proposition (a predicate), parametrized by the day `td` that is *today*. This is because we are interested in being able to reason under the conditions of the past being defined, but would like to vary what "the past" refers to. Similarly, we parametrize the constraints of the paradox so that we are able to reason about whether we are still within them of at each point in the week, which is the level of detail in we are inrested in. This lets us, for example, reason about the impossibility of surprise on Thursday, see Section VII.

We now specify two conditions, implicit in the informal description of the paradox, that are shared across many formal interpretations including [] [], and ones discussed here :

(i) if today is Friday, ie. the whole week has passed, we know that a hanging necessarily happened

```
td = someWeekDay friday
  → ∃ d, hangingOnDay d
```

(ii) if a hanging has occured, it is unique

```
(∃ d, hangingOnDay d)
  → uniqueHanging dayBefore
```

where `uniqueHanging` is a predicate formalizing that after a given day `td`, there can be at most one day on which a hanging occurs, and thus, `uniqueHanging dayBefore` states this about the entire week.

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

We note here that the preconditions in both implications are such that if they are satisfied, intuitively surprise should not be possible : (i) requires that the week is already over, and (ii) requires that a hanging has happened. The time at which we may be surprised by a hanging is strictly before it happens and before the week is over, so neither of these should actually play any role in the reasoning about surprise, which is reflected in our upcoming definition of surprise. However, (i) and (ii) together express all the other constraints of the paradox besides the one requiring the hanging to be a surprise (now we just need to add that part!).

Surprise requires that a future hanging be possible - on more than zero of the remaining weekdays after today. So, the constraint that at least one future possible hanging day exists is clearly necessary. However, the case for at least one possible day not being sufficient to define surprise is a bit more nuanced.

IV. At least one possible day

.

We specify this interpretation of the conditions of the paradox for each possible day as:

```
Definition onePossiblePRDX (td :
    weekAndBefore) :=
  (td = someWeekDay friday
    → ∃ d, hangingOnDay d)
  ∧
  ((∃ d, hangingOnDay d)
    → uniqueHanging dayBefore)
  ∧
  (td < friday ∧ noHangingYet td) ->
    exists d, td < d ∧ ¬ ¬  hangingOnDay d).
```

where the first two conjuncts are as described above, and the third one corresponds to "if today is not yet Friday, there is a possible day on which a hanging may happen", which we refer to as the `onePossible` definition of surprise.

This definition is one of the alternatives discussed in []. 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 ¬ ¬ `hangingOnDay friday`, from which we are not able to deduce that `hangingOnDay friday`.

Note here that including the constraint that *a hanging must happen by the end of the week* is still not sufficient to conclude certainty from unique possibility. The reason for this is that the conclusion that a hanging definitely happened on one of the week days can only be made after Friday has come. The possibility of surprise, meanwhile, has the precondition that the whole week has not passed yet (ie. Friday has not yet come). These preconditions are mutually exclusive.

The condition that if a hanging has happened, it must be unique, also does not give us any reasoning power, as we are not able to conclude `hangingOnDay friday`.

Let us consider what happens if we impose an additional constraint stating that, assuming that (i) there is a possible hanging day, and (ii) a hanging must be unique if it has happened, exactly one *possible* hanging day implies that it *necessarily happens* on that day. This is also explored in [], with a similar conclusion to the one we draw here. The following proposition states that there is a possible hanging day, and that a possible unique day implies certainty of hanging on that day :

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

This proposition, without justification, is the crux of the reasoning the prisoner uses to informally to arrive at the judgement that if a hanging hasn't happened by Thursday, it must happen on Friday. Note here that the inductive reasoning in which the prisoner engages to conclude that the hanging cannot ever be a surprise is, in some sense, superfluous — we can use constructive logic reasoning to prove, without induction, that "if we can conclude from existence plus uniqueness of a possible hanging day, that it is certain on that day, *our judgement about hanging occuring on any day must necessarily be classical*". The proof makes use of the fact that the equality comparison `d = d'` is classical.

The intuition seems to be correct in making the assumption that a unique possibility implies certainty, and the logic leading

to this inconsistency with the informal definition appears solid. However, we will see that the problem lies in the attempt to allow the possibility of, and draw conclusions from, having a *unique possible* hanging day.

This definition leaves us with the following conclusions about defining surprise as having at least one possible hanging day :

(i) such a definition of surprise is not strong enough to allow us to conclude `existsUniqueHappens`, in particular, that a hanging must happen Friday given that it has not occured by Thursday, and is therefore not a surprise in that case ; and

(ii) if we *were* to be able to conclude `existsUniqueHappens`, reasoning about surprise becomes classical, so we can always figure out the hanging day in advance.

Both possibilities appear problematic : (i) does not allow us to make a conclusion that we would like make, and (ii) does not support reasoning non-classically, which removes any ambiguity about the future hanging day, and therefore, the possibility of surprise. Let us see why adding a different additional constraint (ie. other than deriving certainty from a unique possibility) will help.

V. Full paradox : at least two possible days

.

The way we propose to strengthen the conditions of surprise is by increasing the number of possible days required for surprise to two :

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

We can read this as follows :

> There exist two distinct days after the day `td` such that a hanging is possible (ie. cannot disprove that it happens then)

Intuitively, this constraint makes sense — if I am not sure what day something will occur, there must be at least two possible future days on which it could occur, as expressed in `twoPossible`. To formulate the complete conditions of the paradox for each day, we formalize that, in addition to the two constraints justified earlier, we include the constraint that "if today is before Friday, and no hanging has yet happened, there are at least two possible distinct hanging days in the future" :

```
Definition twoPossiblePRDX
    (td : weekAndBefore) :=
  (td = someWeekDay friday
    → ∃ d, hangingOnDay d)
  ∧
  ((∃ d, hangingOnDay d)
    → uniqueHanging dayBefore)
  ∧
  (td < friday ∧ noHangingYet td) ->
    twoPossible td.
```

Now, the first thing we can formally conclude about this definition is that it indeed rules out a Friday hanging. The following lemma says that it is not possible that by Thursday, no hanging has happened, but there are still two distinct possible days for it to happen in the future.

```
Lemma cantBeSurpFriday :
  twoPossiblePRDX (someWeekDay thursday)
  → False.
```

The proof (see code) is trivial, since there is only one day (Friday) left in the week, and no hangings are possible on past days.

This reasoning does not work, however, for any day before Thursday, since there are (for any day before Thursday) at least two future days about which we have no data contradicting the possibility of a hanging. Inductive reasoning in attempt to conclude that a Thursday hanging is predictable on Wednesday does not work because we do not have enough data on Wendesday 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*.

So, this definition of surprise aligns with our intuition by making surprise by a Friday hanging possible, but does not appear to contradict the possibility on any other day. It has, however, an unintuitive feature.

VI. Negation of uniqueness.

With this definition of surprise, surprise is never possible when there is exactly one possible hanging day. The constraint, `uniqueHanging`, that a (provable) hanging day is unique is, in fact, equivalent to the constraint that the (possible) hanging day must be unique,

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

We parametrize the above propositions by td to express that they apply to a specific subset of the week days — those days that are after today, eg., `uniqueMaybe dayBefore` specifies that if a hanging is provable to be on *any two weekdays*, those days must be the same. We prove that they the two propositions are equivalent :

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

The proof of this relies on, again, the fact that $d = d'$ is a classical comparison. Moreover, the `twoPossible` constraint in our definition of surprise *is the negation of* `uniqueHanging` (and therefore also `uniqueMaybe`) :

```
Lemma twoNotUnique :
```

```
∀ td,
¬ uniqueHanging td
↔
twoPossible td.
```

We make an observations about this : a future possible day of the hanging is necessarily not unique. This seems wrong — we would expect a hanging to be unique, it's implicit in the description of the paradox. Let us inspect this closer, however. Our conditions of this paradox on day td are that either the week is over and the hanging has occured, or, for $td \neq thrusday, friday$ surprise is only possible if the hanging has not yet happened. We not that

Once a hanging occurs, we can no longer reason about it having been a surprise That is, a judgement of surprise can only made about an event in the context of a lack of information about it (in this case, whether a hanging will happen on a specific future day).

This is perhaps the most paradoxical idea here. Any reasoning that includes in its hypotheses both that a hanging happened after `td` and that it is a surprise (in our case, specified as `twoPossible td`) is intuitively meaningless. This is not a formal statement, but rather a statement about what surprise means. You cannot be surprised by an event that has already occured.

As a sidenote, this is a nuance of how we use language. One can reason about a surprising past event from the perspective of a time prior to that event, but to do so, the available hypotheses must only include information known at that time, ie. not the occurence of that event. This is why statements such as "I was surprised by a hanging when it happened", or "last week I was surprised when I found out that it had happened to my friend two weeks ago", make sense, and are usually what is meant when one says "I am surprised by a past hanging".

Now, back to why it is ok for hanging day to not to be required to be unique before the hanging occurs : once the (first) hanging occurs on some `d`, we now have `¬ noHangingYet (someWeekDay d)`. That means that the `twoPossible (someWeekDay d)` implication of `noHangingYet (someWeekDay d)` is no longer required to hold, and also that `exists d, hangingOnDay d`, so its consequence `uniqueHanging dayBefore` is now required to hold. So, uniqueness constraint "kicks in" after the hanging first happens.

VII. TGI Thursday

With this definition of the paradox, surprise is already not possible on Thursday. The reason for this is not that the definition allows us to conclude that a hanging will definitely happen on Friday once we get to Thursday. Rather, it is that the conditions for surprise are not satisfied on Thursday in another way (a missing a possible day).

We have already discussed that, since the definition is parametrized by the day on which we judge whether the paradox conditions are met, surprise (and all other conditions of the paradox) actually form a different proposition for each day. And so, an impossibility of surprise Thursday does not affect what we can prove about the rest of the days. With that in mind, we can choose to make a special case for the paradox characterization, which only applies when exactly one day remains as a possibility for a hanging — and there is only one such day, Thursday.

The special case to add is exactly the conclusion we rejected in the `onePossible` version of surprise : `existsUniqueHappens`. Note here that the difference between one- and two-possible versions of surprise, with respect to adding this constraint, is that in the two-case, we use the paradox definition to conclude that Thursday does not allow surprise. In the one-case, on the other hand, we *rely* on the reasoning in `existsUniqueHappens` to conclude surprise is not possible on Thursday, which leads an unsavoury conclusion.

We do not add it, after all, even though the reasoning it allows us to do appears coherent. The reason for this is that the preconditions of this constraint already conflict with those of the `twoPossible` surprise definition, as well as with the preconditions of the universal paradox constraints. So, it does not add additional reasoning power to our definition.

VIII. The week is done

In both versions of the definition of surprise (`twoPossiblePRDX` and `onePossiblePRDX`), Friday is explicitly excluded from the surprise part of the paradox definition. We exclude this day since Friday arriving means that the week has passed and there is no possible way for a hanging to occur. We also conclude that the hanging must have already occured.

IX. Conclusion and Future Work

We have presented a formalization of the unexpected hanging paradox that appears to capture the conditions of the paradox, meanwhile both aligning with the intuition about Friday, and not reasoning ourselves out of the hanging entirely. We formalized this definition, along with some related proofs, in the Coq proof assistant. A few key ideas were needed to achieve this.

The starting point was the observation that constructive logic is required to represent knowledge in order to allow for the possibility that we may not know something to be either true or false — such as whether a hanging occured on a given day. Next, we observe that surprise is a different notion on each of the days, so our definition is parametrized by the day on which we reason about its possibility. Then, we notice that the universal conditions of the paradox only apply when surprise is intuitively not possible (ie. the end of the week or after the hanging), and we separate these out from the surprise specification.

Finally, we formalize and compare two definitions of surprise. The first, `onePossible`, has been exlored in prior work. We discuss why it has consequences contradictory to our intuition about the paradox, and attempts to rectify them in the intuitive way result in the exact undesirable conclusion that the prisoner makes in the paradox description. The second is `twoPossible`, which we proposed, and it is a strengthening of the first definition that does not cause the

reasoning about the future hanging to become classical (unlike `onePossible`).

The surprise formalization we propose drops (in fact, negates) the constraint guaranteeing the uniqueness of the future hanging. The subtlety here is in the idea that the constraint still applies *once the hanging happens*, but multiple distinct days for a future hanging are still required up until then. Our Coq formalization of the paradox stands out from existing work in (i) its mechanization of the problem (ii) providing a resolution to the paradox that aligns with intuition (iii) leading us to making new subtle observations about the constraints of the problem and how they are hidden by language.

As part of future work, a proof of the existence of `hangingOnDay`, such that a version of `twoPossiblePRDX td` (parametrized by the hanging decision function) holds for all days except Thursday,

```
Proposition existsHangFunc :
  ∃ hangingOn, ∀ td,
  twoPossiblePRDX_param
    hangingOn td.
```

would valuable in establishing our formalization as the universally accepted one. Additionally, 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 [] as well as arbitrary elements [].

Another future direction we consider is generalizing the surprise hanging approach we propose to using constructive logic for describing and proving the existence of a function `myPick` which represents choosing an arbitrary value from a decidable set. In particular, given a decidable set `W`,

> For any subset `S ⊆ W` of cardinality at least 2, such that for all `s ∈ W - S`, `¬myPick s`, there exist at least two distinct elements in `S`, such that `¬ ¬ myPick s`.

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.