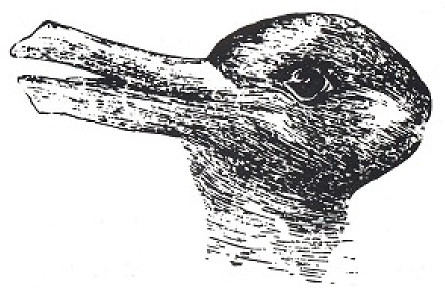
The Meaning of Consistency

by Sven Nilsen, 2018

In this paper I show that a powerful proof system that proves a term to be consistent if and only if the term can be proven to be `true` is relying on a false assumption. Instead, if a term can be proven to be `true`, it only equals consistency if the term is not assumed to be `false`. This means that the truth value of consistency depends on the assumption, which requires a dependently typed variable.



When I look for a duck, I see a duck but no rabbit, and when I look for a rabbit, I see a rabbit but no duck.
This example illustrates what consistency means when it is not obvious what the picture shows.

The meaning of consistency has the following solutions:

Provability/Assumption	**A : true**	**¬A : true**
⊥	A : consistent	A : ¬consistent
A : false	A : ¬consistent	A : consistent
A : true	A : consistent	A : ¬consistent

The law of excluded middle constructs a trivial proof of consistency from some assumption. However, this technique is only valid if and only if neither `A : true` nor `¬A : true` can be proven from no assumption. This shows that the sub-type `consistent` is actually a dependent function of type:

consistent : term(x : Assumption) → [truth(x : Assumption)] true

Assumption := { no_assumption, term_is_false, term_is_true }

truth : Assumption → bool → bool
truth(no_assumption) := \(x : bool) = x <=> $id_{\{bool\}}$
truth(term_is_false) = \(x : bool) = ¬x <=> not
truth(term_is_true) = \(x : bool) = x <=> $id_{\{bool\}}$

true : consistent(no_assumption)	false : ¬consistent(no_assumption)
true : ¬consistent(term_is_false)	false : consistent(term_is_false)
true : consistent(term_is_true)	false : ¬consistent(term_is_true)

The “left side” is proved by the proof system, done for each term, making separate assumptions.

I demonstrate this technique on “This sentence is true”, which is a self-referential sentence:

Assume a human proof system based on intuition.

“This sentence is true” is a sentence that claims to be true.
With other words, it makes an assumption that it is true.
However, no sentence is allowed to inject assumptions into the proof system.
Therefore, the proof system is permitted to test whether the claim is consistent.

First, one has to know whether the sentence can be proven to be `true` or `false`
without making any assumptions. One way to do this is to normalize the sentence.
By normalizing the sentence it becomes “This sentence is either true or false, but not both”.

(true $\veebar$ false) : consistent(no_assumption)	
true : consistent(no_assumption)	(true $\veebar$ false) => true
true	true : consistent(no_assumption) => true

This means the sentence is consistent under no assumption.

Now, assume the sentence “This sentence is true” is false.
One way to do this is to add the assumption to the sentence.
By adding the assumption the sentence it becomes “This sentence is true and false”.

(true $\wedge$ false) : consistent(term_is_false)	
false : consistent(term_is_false)	(true $\wedge$ false) => false
true	false : consistent(term_is_false) => true

This means the sentence is consistent under the assumption the sentence is false.

Now, assume the sentence “This sentence is true” is true.
One way to do this is to add the assumption to the sentence.
By adding the assumption to the sentence it becomes “This sentence is true and true”.

(true $\wedge$ true) : consistent(term_is_true)	
true : consistent(term_is_true)	(true $\wedge$ true) => true
true	true : consistent(term_is_true) => true

Since the sentence is consistent under all assumptions,
the sentence is consistent.

I demonstrate this technique on “This sentence is false”, which is a self-referential sentence:

Assume a human proof system based on intuition.

“This sentence is false” is a sentence that claims to be false.
With other words, it makes an assumption that it is false.
However, no sentence is allowed to inject assumptions into the proof system.
Therefore, the proof system is permitted to test whether the claim is consistent.

First, one has to know whether the sentence can be proven to be \`true\` or \`false\`
without making any assumptions. One way to do this is to normalize the sentence.
By normalizing the sentence it becomes “This sentence is either false or true, but not both”.

(false ⊻ true) : consistent(no_assumption)	
true : consistent(no_assumption)	(false ⊻ true) => true
true	true : consistent(no_assumption) => true

This means the sentence is consistent under no assumption.

Now, assume the sentence “This sentence is false” is false.
One way to do this is to add the assumption to the sentence.
By adding the assumption the sentence it becomes “This sentence is false and false”.

(false ∧ false) : consistent(term_is_false)	
false : consistent(term_is_false)	(false ∧ false) => false
true	false : consistent(term_is_false) => true

This means the sentence is consistent under the assumption the sentence is false.

Now, assume the sentence “This sentence is false” is true.
One way to do this is to add the assumption to the sentence.
By adding the assumption to the sentence it becomes “This sentence is false and true”.

(false ∧ true) : consistent(term_is_true)	
false : consistent(term_is_true)	(false ∧ true) => false
false	false : consistent(term_is_true) => false

Since the sentence is inconsistent under the assumption that it is true,
the sentence is inconsistent.

From the two example on previous pages, one can infer that for every paradox of the kind:

> "This is false"

There exists a consistency tautological statement:

> "This is true"

A consistency tautological statement is `true` both when it assumed to be false and true. However, under no assumption, the truth value of the sentence can not be determined.

Now, one can interpret Gödel's second incompleteness theorem the following way:

> No axiomatic system that includes Peano arithmetic can prove its own consistency.
>
> The interpretation of consistency used in Gödel's second incompleteness theorem:
> The syntactic definition states a theory T is consistent if and only if there is no formula φ such that both φ and its negation $\neg\varphi$ are elements of the set T.
>
> However, the correct interpretation of consistency permits φ and its negation $\neg\varphi$ as elements of the same set, but under an assumption as dependently typed variable.
>
> Transforming the interpretation into correct form:
> The syntactic definition states a theory T(a) is consistent if and only if there is no formula $\varphi(a)$ such that both $\varphi(a)$ and its negation $\neg\varphi(a)$ are elements of the set T(a).
>
> $\gamma := \backslash(\forall\ a,\ \varphi(a) : T(a)\ \{\neg\varphi(a) \wedge \varphi(a)\}) = \varphi(a)$
> $\gamma \wedge \neg\gamma : \exists\ a\ \{[:]\ T(a)\}$ $\quad\quad$ $\gamma \Rightarrow \varphi,\ \exists\ a\ \{[:]\ T(a)\} \Rightarrow T$
> $\varphi \wedge \neg\varphi : T$ $\quad\quad$ The new interpretation implies the old interpretation
>
> In the proof above, the assumption `a` is interpreted abstractly and as the same for the theory.
> This can be improved by separating the assumption of the theory from the formulas.
> Assume a very simple theory with only one consistent formula:
>
> $\forall\ a\ \{\ T(a) : consistent(a) \Rightarrow \forall\ b\ \{\ \varphi(b) : consistent(b)\ \}\ \}$
> $\forall\ a\ \{\ T(a) : consistent(a) \Rightarrow true\ \}$
> $\forall\ a\ \{\ \neg(T(a) : consistent(a)) \vee true\ \}$
> $\forall\ a\ \{\ true\ \}$
> true
>
> One can see from this that if all formulas of a theory are consistent,
> then it does not matter what assumptions we make for the whole theory.
>
> The theory is consistent if and only if all formulas are consistent.
> Therefore, if the theory is inconsistent, there must be some formula that is inconsistent.
> Our definition is decidable for decidable theories, which means consistent theories
> that includes Peano arithmetic must be undecidable on the "left side" for some term.
>
> Under the assumption that the "left side" is decidable, a consistent theory can be proven.