Introduction to the Core Axiom

By Sven Nilsen, 2022

Why developing a Core Axiom?

- Use of symbols is not tautological
- Previous attempts to formalize use of symbols were based on the logical fallacy that nothing should be added to logic, by “explaining” use of symbols without further assumptions
- Path Semantics simply does the groundwork properly, learning from other people’s mistakes

Design Requirements

- Propagate equality without collapsing semantical space of propositions
- Allow reasoning about uniqueness
- Safe local reasoning with unrestricted freedom
- Safe for any two propositions involved in equality

Help to understand the approach

- Propositional Logic was invented on Earth, now it is time to venture out into space – leave old ideas behind
- Less is more: Aim for the smallest theory possible that satisfies design requirements
- Do not use First Order Logic – Do not appeal to general need for reasoning e.g. about time – Just solve the problem

Exploring with humility

- Mathematics is vastly greater than humans or even super-intelligent computers will ever be able to comprehend fully
- Do not “simplify” mathematics, let it be the wild beast it is in its natural environment and follow its trail, learn from it – not your own biases
- There will be time to address general needs – later

Propagating equality

- Equality `==` is the most important operator in all of mathematics
- Use of symbols is basically how to “cause” one equality from another equality – safely
- Equality of symbols implies equality of meaning
- Do not collapse symbols into truth or falsehood – this collapses the semantical space of propositions

Reasoning about uniqueness 1 of 2

- If `a => b` and `a => c`, then if `a` is true, both `b` and `c` are true
- Truth has uniqueness property by implication, since it makes the theorems true from the assumptions – bringing them together into one
- However, we can not use truth – too strong

Reasoning about uniqueness 2 of 2

- If `a => b` and `c => d`, then `a` being true is the only way to tautologically make `b` true. The same holds for `c` to make `d` true. This is a problem, because one wants independence between `a` and `c` when they are symbols
- Truth causes the collapse of the semantical space of propositions into a “cartoonish” representation that people often mistake for the actual semantics

Safe local reasoning

- Since symbols are not tautologies, we need to separate their usage from the general use of logic, such that general logic keeps working
- The core axiom divides propositions into levels where each level can have an indefinite number of propositions
- Use of symbols is about how reasoning in one level affects the reasoning in next level
- These levels are called “Path Semantical Levels”

Safe for any two propositions

- The proposition `a == a` is tautological, which is unsafe because tautologies can not be used to prove anything
- This tautological property of equality is called “reflexivity”
- By lifting equality to a new operator `~~` one can remove reflexivity and get partial equivalence
- The name is “quality” without the “**e**” in “**e**quality”

How to construct quality 1 of 3

- Quality is not possible to express in logic using the normal 16 binary operators, so how do we construct it? How can we make the unthinkable – thinkable?
- The trick is to make `a ~~ a` *undefined*, since it already is undefined by definition when removing reflexivity from equality, one can leverage this explicitly
- So, `a ~~ a` means literally “any proposition”

How to construct quality 2 of 3

- We introduce a qubit operator `~` which is like the unary analogue of quality

  (a ~~ a) == ~a

- In the classical model, `~` uses the argument as a random seed to represent “any proposition”

How to construct quality 3 of 3

- We can now define quality:

    (a ~~ b) == ((a == b) & ~a & ~b)

- The quality and qubit operators are equivalent in the way that both can be defined in terms of each other

Help to understand quality

- Quality is like building a spaceship for leaving Earth – but two more things are needed: Rocket fuel (lifting from equality) and a planned orbit (core axiom)
- Quality also opens up for homotopy levels of logic, but going into depths about homotopy levels is outside the scope of this introduction to the core axiom
- We only need to apply qubit `~` once for quality (homotopy level 2), which means quality is the most important operator among the 4 294 967 296 binary operators in same level

Lifting from equality 1 of 4

- When `a == b` and one can prove that this is an intentional theory, which means neither a tautology nor a paradox, there is a seemingly contradiction in that `a == b` yet they are also different in some sense
- This “same, but different” is possible to formalize precisely using Exponential Propositions which adds a `^` operator to constructive logic
- For example `a ^ b` means `a` is provable from `b`

Lifting from equality 2 of 4

- The expression `a ^ b` seems very similar to `b => a`, but there is an important difference: The `^` operator can not capture variables in the type theory, unlike `=>`, so one can not assume anything else to “force” it to become true – incorruptible in some sense
- This is not Modal Logic – too strong – instead we abandon it in favor of uniform involution

Lifting from equality 3 of 4

- We can reintroduce Modal Logic later using the theory of Avatar Extensions, in order to make `!` (not) uniform in one way and non-uniform in another way, but let us save these technical details for another time
- The most important thing here is that `^` is needed to express precisely how to lift `a == b` into `a ~~ b`

Lifting from equality 4 of 4

- Tautology: `a^true`
- Paradox: `false^a`
- Uniform: `a^true | false^a`
- Theory: `!(a^true | false^a)`
- Therefore, to lift equality into quality, we use:

  (a == b) & !((a == b)^true | false^(a == b)) => (a ~~ b)

Help to understand lifting equality

- Just like rocket fuel is able to lift a huge spaceship from the ground on Earth, lifting equality makes it possible to move beyond general logic into the next homotopy level 2
- We need homotopy level 2 to transfer equality from one world of level 1 into another world of level 1
- However, we do not have a “planned orbit” yet! You can not go to space by aiming the spaceship in a random direction, so we need the “core axiom” for aiming logic

Finally! The Core Axiom

- The Core Axiom of Path Semantics is:

```
((a1 ~~ b1) & (a1 => a2) & (b1 => b2)
 & (a1 < a2) & (b1 < b2))
=> (a2 ~~ b2)
```

- This axiom satisfies all the design requirements

Design Requirements Satisfied

- Propagates equality without collapsing semantical space of propositions
- Allows reasoning about uniqueness
- Safe local reasoning with unrestricted freedom
- Safe for any two propositions involved in equality

Example

- `a == b` implies `c == d` in the next Path Semantical Level when `a => c` and `b => d`, without `a` or `b` needing to be “true” in some sense – but they need to be “same, but different” – just like symbols!
- The propositions `c` and `d` “lives” in the next level and not the same level as `a` and `b`
- The propositions `a` and `b` are unaffected by the reasoning in the next level – unless there is “time travel”

Help to understand the Core Axiom

- Symbols are only useful when they are found in different places, like the letter “a” in one place repeated as “a” in another place
- This is what we mean by “same, but different”
- When we recognize two symbols as the same, we understand – by thinking – what is meant using knowledge of each symbol in their own place
- This is why we need another level, separated from recognition, to make room for reasoning about meaning

How the Core Axiom was designed

- The Core Axiom was constructed like most computer programs, but by creating it deliberately to satisfy the design requirements, testing it and iterate over time
- A lot of work went into understanding the mathematics behind it, because this knowledge could not be learned directly from other people

What can the Core Axiom prove?

- The Core Axiom can prove many intuitive properties of Type Theory – such as:
  - if `a : T, b : U, T != U`, then `a != b`
  - here, members are in level 0, types are in level 1
- The Index Theorem – which implies that when pairing up pieces of structures to natural numbers one can prove isomorphisms

What is the utility of the Core Axiom?

- Current foundations of mathematics use large leaps in thinking – in comparison – which assumes too much about what mathematics can be, e.g. that it needs to be based on Sets or Types. Why not e.g. Time or Randomness?
- The Core Axiom helps us to see what mathematics can be in higher dimensions by making even fewer assumptions about the theory – but also enough to not be tautological in general logic in a safe and clean way

What is the future of mathematics?

- Computer software will become more and more important in the foundations of mathematics, since humans alone do not have the mental capacity to reason about higher homotopy levels
- A very intelligent person can reason about 16 operators, but not 2^32 (over 4 billions) or more
- Balanced focus on theorem proving and mathematical language design