Formalizing Climate Change Mitigation

by Sven Nilsen, 2019

A lot of confusion about climate change mitigation is caused by complex words, unrealistic model parameters and wishful thinking. Resolving this confusion is extremely important, since decisive political action and international collaboration requires a consistent way of reasoning. Recently it was discovered that path semantics is expressive enough to reason about policy problems. In this paper I formalize the most basic language of climate change mitigation in path semantics, such that future work might build on this core and make use of custom automated theorem provers to assist politicians.

Path semantics is a flexible and useful framework for modeling and understanding mathematical languages. It is strictly more expressive than dependently typed systems, but more rigorous than natural languages inspired by logic, such as Lojban. The combination of expressiveness and rigor makes it an efficient tool to build bridges between complex ideas and computable algorithms.

The language of path semantics is in the form of mathematical functions. When some functions form a mathematical relationship relative to each other that is studied in the most general case, it is called a "path". Hence the name "path semantics". The most common paths are:

- Normal paths (they predict something about a function)
- Existential paths (they predict something about the codomain of a function)
- Probabilistic versions of normal and existential paths (they predict probability distributions)

When a problem is formalized in path semantics, it is written as functions that might be fully defined or partially known. The ability to talk about partially known functions makes it an efficient tool to reason about the past and the future.

By formalizing the language of climate change mitigation as functions, the properties of these functions can be used in custom automated theorem provers to reason about climate change mitigation.

The most important concept in climate change mitigation is a "forecasting function". A forecasting function is a function that predicts some part of the future when fully specified. In general, they are assumed to be normal paths of the form:

reality{realistic_assumptions}[assumptions_generator → data_generator] => f

f : A → B
assumptions_generator : assumptions → A
data_generator : data → B
reality : assumptions → data
realistic_assumptions : assumptions → bool

Here, \`f\` represents some forecasting function. The types \`A\` and \`B\` are generic.

When a forecasting function is formalized this way, it is thought of as "grounded". It means that its semantics can be interpreted relative to some imaginary function called \`reality\` that produces data.

All statements in the language of climate change mitigation are implicit statements about `reality`. This function refers to some subset of reality in the more broader sense, making it relevant for real world decision making.

Reality can be thought of as a function that takes assumptions, used to interpret evidence, and produces data. The data is of a form such that `data_generator` maps to types that can be used to compare statements about reality under different assumptions.

For example, when data contains information about e.g. energy, the meaning of such statements are the same for the entire set of realistic assumptions. Usually, the correct assumptions to make about reality are unknown. Indirect measurements might change depending on which assumptions one makes. Assumptions are therefore choices of how to interpret evidence. The data produced is according to the formal definition of functions that specify how direct measurements refers to indirect measurements.

Reality is not thought of as a function of time, because many useful statements can be made about reality that are not statements about time.

Instead, when there is a distribution over time, for example greenhouse gas emissions:

    ghg_emissions : year → Gt

    ghg = greenhouse gas
    Gt = gigatons

The `data_generator` function maps from data to a function:

    data_generator : data → (year → Gt)

Where the `assumptions_generator` has the type:

    assumptions_generator : assumptions → political_action

The forecasting function has then the type:

    ghg_emissions_scenario : political_action → (year → Gt)

Putting all this together, one gets:

    reality{realistic_assumptions}[assumptions_generator → data_generator] => ghg_emissions_scenario

    ghg_emissions_scenario : political_action → (year → Gt)
    assumptions_generator : assumptions → political_action
    data_generator : data → (year → Gt)
    reality : assumptions → data
    realistic_assumptions : assumptions → bool

In general, it is sufficient to specify the type of the forecasting function.

The separation of assumptions and data makes it easier to reason about climate change mitigation. It permits different thought experiments and hypothetical scenarios, while being concrete enough to make use of measured data. This formalization makes it also possible to keep track of real world data.

A statement can be made about the forecasting function, for example:

```
∀ pa : political_action, y : (>= 2019) {
        ghg_emissions_scenario(pa)(y) >= ghg_emissions_scenario(pa)(2019)
}
```

This means “in all years including 2019 and following years, there will be greenhouse gas emissions equal or larger than in the year 2019”. It is just an example statement, which might be `true` or `false`.

One can also replace `political_action` as a quantified variable with `ghg_emissions` as a collection:

```
ghg_emissions := sift pa : political_action { ghg_emissions_scenario(pa) }

ghg_emissions : [year → Gt]
```

The same statement about the forecasting function can now be written in a new way:

```
∀ f : ghg_emissions, y : (>= 2019) { f(y) >= f(2019) }
```

By abstracting away the assumptions (only require them to be somehow realistic), one can spot general patterns and trend in the forecasting prediction, that are of vital importance for political decisions.

Only realistic assumptions are relevant. When something holds for all realistic assumptions, it might be regarded as a solid future prediction. If one relies on some assumption that violates solid future predictions, then one relies on a very likely false assumption and the reasoning becomes inconsistent.

The point of using a formalized language is to build custom tools that verify the correctness of assumptions for future decision making. A major motivation is to avoid confusion, since indecisive political action on climate change mitigation might potentially cause the deaths of millions, maybe hundreds of millions of people, or perhaps even risk the extinction of humanity.

There is no time to make huge mistakes. All political action must be done in the context of realistic assumptions, because actions made under idealized or false assumptions will be likely less effective. In the worst case scenario, false assumptions might lead to insufficient resources for vital political action.

On the other hand, decisions based on the entire range of realistic assumptions are likely guaranteed to be effective. This is the best effort politicians can make to save lives. Instead of assuming that some assumptions holds, which later might turn out to be false, one should look at the general trends and patterns and decide the best course of action across all plausible scenarios.

For example, the goal of The Paris Agreement is to limit the average temperature above pre-industrial levels to 2 degrees Celcius. This might turn out to be unrealistic, so if politicans assume this to be true, they might make a series of decisions that do not save lives in the realistic scenario. It could cause the deaths of millions, maybe even hundreds of millions of people, just because of not taking consistent reasoning seriously. Formalizing climate change mitigation removes such dangerous biases.