Time Interpretation

by Sven Nilsen, 2021

In this paper I introduce a time interpretation of Path Semantical Logic.

One technique to build intuition of Path Semantical Logic[1] is think about layered propositions as time.

The Time Interpretation of Path Semantical Logic builds on these three core ideas:

1. Propositions as events
2. Passage of time as layers of propositions
3. Clocks as intersecting propositions in each layer

The rest of this paper I go through these three code ideas and explain them using examples.

1. Propositions as events

Path Semantical Logic is an extension of Propositional Logic[2] with layers of propositions.
Propositional Logic can be interpreted in different ways. This is what makes it so useful.

The standard interpretation is the following:

a := (a = true)	The proposition \`a\` is assigned the value \`true\`

However, it could also be the existence of some symbol \`a\` in a symbolic free form grammar[3]:

a := ∃ sym : Sentence { sym = “a” }	The symbol \`a\` in a free form grammar

In the standard interpretation, \`a\` just means “\`a\` is true”.
In the grammar interpretation, \`a\` means “\`a\` exists somewhere in the sentence”.

When an interpretation is chosen, it is important to stick with it throughout the entire logic.
The interpretation determines how one “reads” the logical expressions.

Propositions as events means that propositions are interpreted as events happening in time:

a := \`a\` happens

For example:

it_is_nice_weather ∧ it_is_sunny	It is nice weather and it is sunny at the same time

While this distinction might seem trivial at first, it gets more important when combined with layers.
I will go into more depths about this in the next section.

2. Passage of time as layers of propositions

Path Semantical Logic is an extension of Propositional Logic[2] with layers of propositions.
Each layer might be thought of as a moment in time.

The standard way of writing proofs in Path Semantical Logic is by grouping propositions by layers:

(a b c) (d e) (f):	
a => d	\`a\` causes \`d\`
e => f	\`e\` causes \`f\`

For example, the \`(d e)\` group means that \`d\` and \`e\` belong to the same layer.

Since propositions can be interpreted as events, it becomes natural to think of each layer as a moment.

The passage of time goes from left to right, such that:

(a b c)	happens before	(d e)
(d e)	happens before	(f)

One can add as many layers as desired.

Implication \`=>\` is interpreted in the following ways:

- Causation: From one layer to some later layer
- Time travel causation: From one layer to some previous layer
- Structure of being: Within one layer

For example, in the movie “Back to the Future” (1985), Marty McFly travels back in time and accidentally prevents his mother from falling in love with his father, and he must make them fall in love in order to return back to his own future.

(a) (b):	(Marty’s mother giving birth to Marty) (Marty going back in time)
a => b	Marty’s mother giving birth to Marty causes Marty going back in time
b => ¬a	Marty going back in time prevents Marty’s mother giving birth to Marty

It is provable from this that \`¬a\`, meaning that Marty’s mother does not give birth to Marty.
However, it is not possible to prove \`b\` or \`¬b\`. This is \`b\` is not assumed to happen.
Yet, it is possible to prove \`b ∨ ¬b\`, so Marty either exists or he does not exist.
When reasoning about time travel problems, it is often best to not assume concrete events.

What makes Path Semantical Logic different from Propositional Logic is the following:

(a b) (c d)
(a => c) ∧ (b => d)   =>   (a = b) => (c = d)

Propositions that are equal within one layer, causes equality, by causation, in the next layer.

3. Clocks as intersecting propositions in each layer

A clock is a special proposition per layer that is used to synchronise events:

(c1 a b) (c2 c d) (c3 e f):
a ∧ c1 => c
b ∧ c1 => d
c ∧ c2 => e
d ∧ c2 => f

The clock makes reasoning about events possible in the future:

(a = b) ∧ c1 => (c = d)
(c = d) ∧ c2 => (e = f)

When \`a = b\` and the clock ticks one second \`c1\`, \`a\` and \`b\` "becomes" \`c = d\`.
This preserves the structure of being, such that time transitions smoothly from one moment to the next.

Without a clock, the structure of being in the future will collapse:

(a b) (c d) (e f):
a => c
b => d
c => e
d => f

From this it is possible to prove the following:

(a = b) => (c => e) ∧ (e => c)

Notice that the two events \`c\` and \`e\` causes each other, creating a time travel loop.

Without a clock, instead of the structure of being to align up within one layer only, it propagates to all future layers. This means that the future is collapsed to a single moment, instead of consisting of multiple moments following in sequential order.

The clock prevents this collapse from happening by introducing an extra assumption, in order to make events in one layer cause events in the following layer.

References:

[1] “Path Semantical Logic”
AdvancedResearch – Reading Sequence on Path Semantics
https://github.com/advancedresearch/path_semantics/blob/master/sequences.md#path-semantical-logic

[2] “Propositional calculus”
Wikipedia
https://en.wikipedia.org/wiki/Propositional_calculus

[3] “Propositional Logic as Symbolic Free Form Grammar”
Sven Nilsen, 2019
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/propositional-logic-as-symbolic-free-form-grammar.pdf