Propositional Logic Interpretation of Answered Modal Logic

by Sven Nilsen, 2020

In this paper I show that Answered Modal Logic can be interpreted, in part, with Propositional Logic. This model can only be fully developed under Higher Order Operator Overloading.

For `n` variables, the semantic model of canoncial expressions in Answered Modal Logic form an n-dimensional cube, where each dimension represents a variable in a modal set `{!◇, ¬!◇, □}`.

The corners of this n-dimensional cubes naturally can model propositions:

□X	X is true
!◇X	X is false

The generalized Propositional Logic gate NOT is defined as following:

not(!◇X) = □X	¬!◇X = ¬!◇X
not(□X) = !◇X	¬□X = ¬□X
not(¬!◇X) = ¬□X	¬¬!◇X = !◇X
not(¬□X) = ¬!◇X	¬¬□X = □X

This is also consistent with:

not[¬] <=> not

To derive a full Propositional Logic interpretation, it is sufficent to construct a NAND gate.
Since NOT is already defined, the remaining work is to construct a AND gate:

and(!◇X, !◇X) = !◇X	Carries over from Propositional Logic
and(!◇X, ¬!◇X) = !◇X	See proof D (next page)
and(!◇X, ¬□X) = !◇X	See proof A (next page)
and(!◇X, □X) = !◇X	Carries over from Propositional Logic
and(¬!◇X, !◇X) = !◇X	See proof D (next page)
and(¬!◇X, ¬!◇X) = ¬!◇X	See proof C (next page)
and(¬!◇X, ¬□X) = ¬!◇X	See proof A (next page)
and(¬!◇X, □X) = ¬!◇X	See proof B (next page)
and(¬□X, !◇X) = !◇X	See proof A (next page)
and(¬□X, ¬!◇X) = ¬!◇X	See proof A (next page)
and(¬□X, ¬□X) = ¬□X	See proof C (next page)
and(¬□X, □X) = ¬□X	See proof B (next page)
and(□X, !◇X) = !◇X	Carries over from Propositional Logic
and(□X, ¬!◇X) = ¬!◇X	See proof B (next page)
and(□X, ¬□X) = ¬□X	See proof B (next page)
and(□X, □X) = □X	Carries over Propositional Logic

Notice that this is an operator on the functions of a variable `X`,
which is permitted by using Higher Operator Overloading (HOOO) semantics.

The NAND gate is constructed by using `nand <=> not . and`.

For functions of different variables, one can not use the semantics of HOOO.
However, in some cases there exists a model in Propositional Logic.

and(□X, □Y)	Undefined, but has a model in propositional logic (corners of the cube)
and(□X, ¬!◇Y)	Undefined, no model

The proofs of AND are as following.

Since `and` is commutative:

and(!◇X, ¬□X) = and(¬□X, !◇X)
and(□X, ¬□X) = and(¬□X, □X)

The following cases are somewhat intuitive, since `¬□X = {!◇, ¬!◇, □}X`:

and(!◇X, ¬□X) = !◇X	Proof A
and(¬!◇X, ¬□X) = ¬!◇X	

The second case is a bit trickier:

∵	and(□X, ¬□X)	Proof B
∴	and(□X)(¬□X)	
∴	id(¬□X)	Using `and(true)(x) => id(x)`
∴	¬□X	

The same trick can be used here:

and(□X, ¬!◇X) = ¬!◇X

Two trivial cases are the following:

and(¬□X, ¬□X) = ¬□X	Proof C
and(¬!◇X, ¬!◇X) = ¬!◇X	

The only case left (two commutative) is the following, which I define using intuition:

and(!◇X, ¬!◇X) = !◇X	Proof D

Q.E.D.