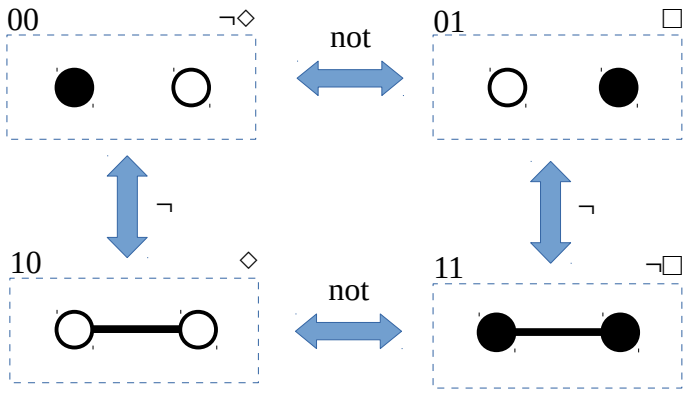
Visualizing Inversion vs NOT in Answered Modal Logic

by Sven Nilsen, 2020

In this paper I visualize inversion (`¬`) vs the generalized logical NOT in Answered Modal Logic.

In Answered Modal Logic, there are two ways to flip bits encoded as a Cubical Binary Code:



The `¬` function is used for inversion, which is the semantics that follows from the answered predicate. In this model, the canonical form of the 3-bit semantic model can be chosen to be symmetric:

100 001 010 111

The `not` function is generalized logical NOT, which is the semantics that follows from the Propositional Logic interpretation of Answered Modal Logic using the `¬□ = [¬◇, □)` interval model:

100 001 011 110

Although the two 3-bit semantic models have different interpretations,
they are compatible using the Cubical Binary Code encoding, which uses only 2 bits:

not(00) = 01	¬(00) = 10	The two functions flip different bits
not(01) = 00	¬(01) = 11	
not(10) = 11	¬(10) = 00	
not(11) = 10	¬(11) = 01	

This idea can be generalized using a higher order `not_n` which has the following property:

∀ x, y { not_n(x)[not_n(y)] <=> not_n(x) }

not_n(0) = not
not_n(1) = ¬