Extracting Bits in Answered Modal Logic

by Sven Nilsen, 2020

In this paper I describe the extraction process of Answered Modal Logic in more detail.

The expression `¬!◇X ∨ □X` can be encoded as following:

011

The first bit tells whether `!◇` belongs to the set.
The second bit tells whether `¬!◇` belongs to the set.
The third bit tells whether `□` belongs to the set.

A special case is the empty expression:

000 `` empty expression

Handling empty expressions properly can be a bit tricky, since there is another special case:

111 ¬□X

When there are multiple variables, one can fill out the unmentioned variables:

¬□X = ¬□X ∧ ¬□Y

111
111
111

The problem is that it is not allowed to remove the last term.
At least one term is needed to witness the fill operation:

¬□X = ¬□Y

This is not equal to an empty expression:

¬□X ¬= `` Notice that `¬=` uses `not`

You can prove that that this rule fills out all the cases:

∵ ¬□X
∴ !◇X ∨ ¬!◇X ∨ □X
∴ (!◇X ∧ ¬□Y) ∨ (¬!◇X ∧ ¬□Y) ∨ (□X ∧ ¬□Y)
∴ (!◇X ∧ (!◇Y ∨ ¬!◇Y ∨ □Y)) ∨ (¬!◇X ∧ (!◇Y ∨ ¬!◇Y ∨ □Y)) ∨ (□X ∧ (!◇Y ∨ ¬!◇Y ∨ □Y))
∴ (!◇X ∧ !◇Y) ∨ (!◇X ∧ ¬!◇Y) ∨ (!◇X ∧ □Y) ∨ (¬!◇X ∧ !◇Y) ∨ (¬!◇X ∧ ¬!◇Y) ∨ (¬!◇X ∧ □Y) ∨ (□X ∧ !◇Y) ∨ (□X ∧ ¬!◇Y) ∨ (□X ∧ □Y)