Recursive Booleans

by Sven Nilsen, 2021

In this paper I introduce a way of extending booleans recursively with new values.

In Recursive Booleans, each new “true” value is defined recursively:

$$\forall\ x : \forall\ i\ n\text{-}1\ \{(!= T_i)\}\ \{\ \mathrm{and}(x, T_n) = x\ \}$$

Each new “false” value is defined recursively:

$$\forall\ x : \forall\ i\ n\text{-}1\ \{(!= F_i)\}\ \{\ \mathrm{and}(x, F_n) = F_n\ \}$$

Notice that the definition only uses AND.
This is because involutions and normal paths “fills out” the rest of the truth tables using assumptions.

The assumptions about involutions and normal paths might vary,
which associates a “family” of many-value logics to some definition of Recursive Booleans.
One common assumption is e.g. that the AND is commutative.

Notice that the definition of a “true” value only depends on previous “true” values.
Similarly, the definition of a “false” value only depends on previous “false” values.

This means, that one can choose to extend in the “true” or “false” directions.
Any extension is defined in terms of a dimensional 2D vector `(<dimension true>, <dimension false>)`.

Each new value is uniquely defined in any given many-value logic.
It means no value can share the same definition as it leads to a contradiction.

(0, 0)	(1, 0) $T_0\ T_0$ => T_0	(2, 0) $T_0\ T_1$ => T_1 $T_1\ T_1$ => T_1	(3, 0) $T_0\ T_2$ => T_2 $T_1\ T_2$ => T_2 $T_2\ T_2$ => T_2
(0, 1) $F_0\ F_0$ => F_0	(1, 1) $F_0\ T_0$ => F_0	(2, 1) $F_0\ T_1$ => F_0	(3, 1) $F_0\ T_2$ => F_0
(0, 2) $F_0\ F_1$ => F_0 $F_1\ F_1$ => F_1	(1, 2) $F_1\ T_0$ => F_1	(2, 2) $F_1\ T_1$ => F_1	(3, 2) $F_1\ T_2$ => F_1
(0, 3) $F_0\ F_2$ => F_0 $F_1\ F_2$ => F_1 $F_2\ F_2$ => F_2	(1, 3) $F_2\ T_0$ => F_2	(2, 3) $F_2\ T_1$ => F_2	(3, 3) $F_2\ T_2$ => F_2

The table above shows new cases in the truth table of AND. Same colors are equivalent.