Set Theory vs Boolean Functions

by Sven Nilsen, 2019

Using Higher Order Operator Overloading[1] (HOOO), one can compare operations in Set Theory[2] with their equivalent operations using Boolean Functions[3]:

	Set Theory	HOOO
Set membership	a ∋ b	b(a)
Union	a ∪ b	a ∨ b
Intersect	a ∩ b	a ∧ b
Exclude	a \ b	a ∧ ¬b
Subset	a ⊆ b	∀ x { (a => b)(x) }
Strict subset	a ⊂ b	∀ x { (a => b)(x) } ∧ ¬∀ x { (b => a)(x) }

Notation:

\`∨\`	Logical gate OR
\`∧\`	Logical gate AND
\`¬\`	Logical gate NOT
\`=>\`	Logical gate IMPLY (material implication)
\`∀\`	For-all loop

From this overview, it is easy to see that "subset" and "strict subset" are computationally expensive.

The strict subset property can be proven from another definition:

∀ x { (a => b)(x) } ∧ ∃ x { (b ∧ ¬a)(x) }
∀ x { (a => b)(x) } ∧ ¬∀ x { ¬(b ∧ ¬a)(x) }
∀ x { (a => b)(x) } ∧ ¬∀ x { (¬b ∨ a)(x) }
∀ x { (a => b)(x) } ∧ ¬∀ x { (b => a)(x) }

References:

[1] “Higher Order Operator Overloading”
Sven Nilsen, 2018
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/higher-order-operator-overloading.pdf

[2] “Set theory”
Wikipedia
https://en.wikipedia.org/wiki/Set_theory

[3] “Boolean function”
Wikipedia
https://en.wikipedia.org/wiki/Boolean_function