Union of Existential Paths

by Sven Nilsen, 2019

In this paper I show a way to take the union of existential paths of boolean functions with logical OR.

The major result of this paper is the following two laws:

$\exists f\{h\} \lor \exists g\{h\} \quad \Leftrightarrow \quad \text{if}(\exists(f \lor g)\{h\}, \exists(f \land g)\{h\})$

$f : T \to \text{bool}$
$g : T \to \text{bool}$
$h : T \to \text{bool}$
$\text{if} : (\text{bool} \to \text{bool}) \times (\text{bool} \to \text{bool}) \to (\text{bool} \to \text{bool})$
$\text{if}(a : \text{bool} \to \text{bool}, b : \text{bool} \to \text{bool}) = \backslash(x : \text{bool}) = \text{if } x \ \{ a(x) \} \text{ else } \{ b(x) \}$

These laws are interpreted using Higher Order Operator Overloading.

In first-order logic there is a "there-exists-loop", also called an "any-loop":

$\exists\ x \ \{ f(x) \} \quad \Leftrightarrow \quad \text{any } x \ \{ f(x) \}$

This loop returns `true` if `f` returns `true` for some input and `false` otherwise:

$\exists\ x \ \{ f(x) \} \quad \Leftrightarrow \quad (\exists f)(\text{true})$

Here, `∃f` means the existential path of `f`.

If the body contains an if-expression filtering by `h`, then one can say the loop iterates over `h`:

$\exists\ x \ \{ \text{if } h(x) \ \{ f(x) \} \text{ else } \{ \text{false} \} \} \quad \Leftrightarrow \quad \exists\ x : h \ \{ f(x) \}$

One can also say that `x` has the sub-type `h`, written `x : h`.

If there are two such loops connected by logical OR, then the two loops can be joined:

$\exists\ x : h \ \{ f(x) \} \lor \exists\ x : h \ \{ g(x) \} \quad \Leftrightarrow \quad \exists\ x : h \ \{ f(x) \lor g(x) \}$

However, the same is not true for logical AND:

$\exists\ x : h \ \{ f(x) \} \land \exists\ x : h \ \{ g(x) \} \quad <\neg=> \quad \exists\ x : h \ \{ f(x) \land g(x) \}$

Intuitively, if two any-loops iterates over the same collection, the performance can be improved by joining the two loops together. If `f` returns `true` for some input and `g` returns `true` for some input, then it is not always the case that `f` returns `true` for some input as when `g` returns `true`.

Previously, I showed that the same law does not work for logical AND. However, if `f` returns `true` for all inputs and `g` returns `true` for all inputs, then they both return `true` for the same input:

$$\forall\ x : h \{ f(x) \} \land \forall\ x : h \{ g(x) \} \quad \Leftrightarrow \quad \forall\ x : h \{ f(x) \land g(x) \}$$

So, there is a similar law for logical AND, but for for-all loops instead of there-exists-loops.

These two laws are related through the existential path of boolean functions.

When the existential path `∃f{h}` of `f{h}` returns `true` for input `true`,
it means there exists some input `x : h` of `f` such that `f` returns `true`:

$$\exists\ x : h \{ f(x) \} \quad \Leftrightarrow \quad (\exists f\{h\})(\text{true}) \qquad (1)$$

When the existential path `∃f{h}` of `f{h}` does not return `true` for `false`,
it means all input `x : h` of `f` makes `f` return `true`:

$$\forall\ x : h \{ f(x) \} \quad \Leftrightarrow \quad \neg(\exists f\{h\})(\text{false}) \qquad (2)$$

Using Higher Order Operator Overloading (HOOO) on the two laws:

$$\exists\ x : h \{ f(x) \lor g(x) \} \quad \Leftrightarrow \quad \exists\ x : h \{ (f \lor g)(x) \}$$
$$\forall\ x : h \{ f(x) \land g(x) \} \quad \Leftrightarrow \quad \forall\ x : h \{ (f \land g)(x) \}$$

In the first case, one can use 1) to prove the following:

$$(\exists f\{h\})(\text{true}) \lor (\exists g\{h\})(\text{true}) \quad \Leftrightarrow \quad (\exists(f \lor g)\{h\})(\text{true})$$

$$\exists\ x : h \{ f(x) \} \lor \exists\ x : h \{ g(x) \} \quad \Leftrightarrow \quad (\exists f\{h\})(\text{true}) \lor (\exists g\{h\})(\text{true})$$
$$\exists\ x : h \{ (f \lor g)(x) \} \quad \Leftrightarrow \quad (\exists(f \lor g)\{h\})(\text{true})$$

In the second case, one can use 2) to prove the following:

$$\neg(\exists f\{h\})(\text{false}) \land \neg(\exists g\{h\})(\text{false}) \quad \Leftrightarrow \quad \neg(\exists(f \land g)\{h\})(\text{false})$$

$$\forall\ x : h \{ f(x) \} \land \forall\ x : h \{ g(x) \} \quad \Leftrightarrow \quad \neg(\exists f\{h\})(\text{false}) \land \neg(\exists g\{h\})(\text{false})$$
$$\forall\ x : h \{ (f \land g)(x) \} \quad \Leftrightarrow \quad \neg(\exists(f \land g)\{h\})(\text{false})$$

Using De Morgan’s law in the second case:

$$\neg(\exists f\{h\})(\text{false}) \land \neg(\exists g\{h\})(\text{false}) \quad \Leftrightarrow \quad \neg(\exists(f \land g)\{h\})(\text{false})$$
$$\neg((\exists f\{h\})(\text{false}) \lor (\exists g\{h\})(\text{false})) \quad \Leftrightarrow \quad \neg(\exists(f \land g)\{h\})(\text{false})$$
$$(\exists f\{h\})(\text{false}) \lor (\exists g\{h\})(\text{false}) \quad \Leftrightarrow \quad (\exists(f \land g)\{h\})(\text{false})$$

Together I have handled the cases when both `f{h}` and `g{h}` returns the same output:

$$(\exists f\{h\})(\text{true}) \lor (\exists g\{h\})(\text{true}) \quad \Leftrightarrow \quad (\exists(f \lor g)\{h\})(\text{true})$$
$$(\exists f\{h\})(\text{false}) \lor (\exists g\{h\})(\text{false}) \quad \Leftrightarrow \quad (\exists(f \land g)\{h\})(\text{false})$$

To combine the cases into a single function, I use the `if` function:

$$\exists f\{h\} \vee \exists g\{h\} \quad \iff \quad \mathrm{if}(\exists(f \vee g)\{h\}, \exists(f \wedge g)\{h\})$$

$$\mathrm{if} : (\mathrm{bool} \to \mathrm{bool}) \times (\mathrm{bool} \to \mathrm{bool}) \to (\mathrm{bool} \to \mathrm{bool})$$
$$\mathrm{if}(a : \mathrm{bool} \to \mathrm{bool},\ b : \mathrm{bool} \to \mathrm{bool}) = \backslash(x : \mathrm{bool}) = \mathrm{if}\ x\ \{\ a(x)\ \}\ \mathrm{else}\ \{\ b(x)\ \}$$

This results in the law that is the major result of this paper.