

## C Semantics

- `mux(AggregateExpression<Boolean>, AggregateExpression<T>, AggregateExpression<T>): AggregateExpression<T>`
- `neighbor(AggregateExpression<T>): AggregateExpression<Map<Int, T>>`
- `branch(AggregateExpression<Boolean>, AggregateExpression<T>, AggregateExpression<T>): AggregateExpression<T>`
- `loop(T, (AggregateExpression<T>) -> AggregateExpression<T>): AggregateExpression<T>`
- `selfID(): AggregateExpression<Int>`

produces

## E Slot

Condition  
Then  
Else  
Neighbor

Key<T>(valure: T)  
Operand(index: Int)

## I

«interface»

*AggregateExpression*

- `compute(List<Slot>, Context): StateFlow<ExportTree<T>>`

emits

uses

## I

«interface»  
*ExportTree*

root: T  
children: Map<Slot, ExportTree<?>>

- `followPath(List<Slot>): ExportTree<?>?`

## I

«interface»  
*Context*

selfID: Int  
sensorsStates: StateFlow<Map<String, ?>>  
neighborsStates: StateFlow<Map<Int, ExportTree<?>>>

- `updateLocalSensor(String, T): Unit`
- `receiveExport(Int, ExportTree<?>): Unit`