Simple reductions

Fabien Coelho

April 30, 2024

1 Imported domains

import reference from "ri.newgen"
import preference from "ri.newgen"
import entity from "ri.newgen"
import statement from "ri.newgen"

2 Operators

Current operators are stored directly. We could add user defined operators if required. None is used to mark dead reductions.

reduction_operator = { none , min , max , sum , csum , prod , and , or , bitwise_and , bitwise_or , bitwise_xor , eqv, neqv}

3 Reduction

A reduction is a reference describing the object on which the reduction is performed. This object may be a scalar, a full array, part of an array, and so on. The operator is also specified. The dependences is the list of entities the reduction reference depends on, and is used in the construction algorithm. The trusted references are those the effects of contribute to the reduction. They are only used and significant in the proper reductions construction algorithm.

reduction = reference x op:reduction_operator x dependences:entity* x trusted:preference*

List of reductions are separated.
reductions = list:reduction*

4 Functions

Function from statement to reductions to be used for proper and cumulated reductions.

pstatement_reductions = persistent statement-¿reductions