

How to build your own analysis: Specifying the framework of your analysis in the PROBLEM section

» [Back to overview](#)

In the PROBLEM section you define the underlying framework for your analysis. You have to specify:

direction	The direction of your analysis, either <i>forward</i> or <i>backward</i>
carrier	The underlying lattice of the analysis. This must be a complete lattice which can be a built-in type of PAG/WWW or a complex type defined in the TYPE section .
init	The initial value at each node. This can be any FULA expression but will usually be the neutral element of the combine function (e.g. <i>bot</i> for <i>lub</i> or <i>top</i> for <i>glb</i>).
init_start	The initial value of all nodes that have no incoming edges. Typically this is the start of the analysis (the <i>begin</i> node of the main program for <i>forward</i> analyses or the <i>end</i> node of the main program for <i>backward</i> analyses). This can be any FULA expression.
combine	The combine function specifies how information from multiple incoming edges is combined at the entry of a node. It can be one of the FULA operators <i>lub</i> (least upper bound) or <i>glb</i> (greatest lower bound), or the name of a FULA function. This function can be one of the predefined functions <i>intersect</i> or <i>union</i> if your carrier is a power set (in fact, these are only alternative names for <i>glb</i> and <i>lub</i> in this case), or any other function defined in the SUPPORT section which has type <code>cType * cType -> cType</code> where <code>cType</code> is the type specified as the carrier.

The following items are optional:

equal	You may specify your own comparison function that checks whether the exit-information at a node has changed. You specify the name of a SUPPORT function which must have type <code>cType * cType -> bool</code> where <code>cType</code> is the type specified as the carrier. The first argument is the old data flow value of the node, the second argument is the new value. If you do not specify an <i>equal</i> function, the default function <code>standard_equal(a,b) = (a=b)</code> will be used.
widening	You may specify a widening function that applies widening if the <i>equal</i> function returns false (i.e. the information at a node has changed). You specify the name of a SUPPORT function which must have type <code>cType * cType -> cType</code> where <code>cType</code> is the type specified as the carrier. The first argument is the old data flow value of the node, the second argument is the new value. If you do not specify a <i>widening</i> function, the default function <code>standard_widen(a,b) = b</code> will be used (this function just returns the new value).
narrowing	You may specify a narrowing function just like a widening function. If you do not specify a <i>narrowing</i> function, the default function <code>standard_narrow(a,b) = b</code> will be used.

Next step

- [The FULA language](#)

Contents

1. [Overview](#)
2. [Specifying datatypes in the TYPE section](#)
3. **Specifying the framework of your analysis in the PROBLEM section**
4. [The FULA language](#)
5. [Global FULA variables](#)
6. [Specifying the TRANSFER section](#)
7. [The SUPPORT section](#)
8. [Built-in PAG/WWW functions](#)
9. [A formal description of the analysis specification language](#)

Search

PAG/WWW  for Search 
