Anubis - Feature #35
Lazy boolean operator |

04/17/2008 05:38 PM - SpiceGuid -

Status:	Rejected	**Start date:**	
Priority:	Low	**Due date:**	
Assignee:	Alain Prouté	**% Done:**	0%
Category:	Virtual Machine	**Estimated time:**	0.00 hour
Target version:	1.x		
Platform:		**Triage Stage:**	
Resolution:	invalid		

Description

From the library source, i guess **Bool x & Bool y** is *lazy* whereas **Bool x | Bool y** is *strict*.

If confirmed then it means the natural coding of an existencial predicate and the natural coding of a universal predicate differ in efficiency.

It seems to me, as a remote consequence of *De Morgan* law, **Bool &** and **Bool |** should be equally efficient.

History

#1 - 04/17/2008 10:35 PM - Alain Prouté

Actually, none is lazy, i.e. all are strict unfortunately. Something is missing such as either the possibility of putting small functions inline (which would do as if the call was lazy), or introducing macros.

For the time being, inline functions are **almost** implemented. I need to have a look at this because maybe no so much work is to be done. As for macros, this project is only for the next boostrapped version.

The reason why boolean & is defined in predefined.anubis is just that it is used in the invisible part of predefined.anubis. But it is defined in the same style as boolean | in basis.anbis, i.e. in Anubis, hence has the same behavior.

#2 - 04/17/2008 10:36 PM - Alain Prouté

- Status changed from New to Assigned

#3 - 04/20/2008 03:40 PM - SpiceGuid -

- Status changed from Assigned to Closed

- Resolution set to invalid

closed. motive: inaccurate statement of the problem and solution.

#4 - 01/20/2009 09:10 AM - Anonymous

- Status changed from Closed to Rejected

- Platform deleted (All)