

4.67! Fundamentally, an infinite loop occurs when the same function gets re-called with no change in loop control, or the loop control can never terminate. (e.g. for $i := 0; i \geq 0; i++$).

The query loop control "pushes" towards binding all variables, in which case, exit the loop successfully. Or find the current unifications lead to a dead end, in which case, exit with a failure/empty result.

Therefore, we need only concern ourselves with the first case: when the same sub-query gets (re-)invoked with the same bindings as the previous time.

Check can be done by unification, perhaps.

Simplest implementation: keep track of sub-query calls along with the binding frame at the time of call, and then work back through history for "last call" to current sub-query to compare bindings: if "last call" exists.

No implementation, since I don't yet know how the query-language is implemented.