lib/basic/nested-eval-problem.ath

```
# This example, given by Kostas in his r1813 log message, works now after
# the checkin of drive.sml and re.sig in r1819.

datatype N := zero | (S N)

module N {
declare Plus: [N N] -> N

module Plus {
assert right-zero    := (forall ?n . ?n Plus zero = ?n)
assert right-nonzero := (forall ?m ?n . ?n Plus S ?m = S (?n Plus ?m))
}
declare Times: [N N] -> N
module Times {
assert right-zero    := (forall ?x . ?x Times zero = zero)
assert right-nonzero := (forall ?x ?y . ?x Times S ?y = (?x Times ?y) Plus ?x)
(print "\n3 Plus 3 = " (eval (Plus (S S S zero) (S S S zero))))
}
}
```