Ruby - Bug #3116

block call with `do' keyword in `until' condition causes parse error

04/09/2010 11:13 PM - mame (Yusuke Endoh)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:	1.9.2		
ruby -v:	ruby 1.9.2dev (2010-04-09 trunk 27030) [i686-linux]	**Backport:**	

Description

=begin
(derived from #1240)

I noticed that block call with do' keyword does not work in until' condition:

until begin 1.times { }   end do end # ok
until begin 1.times do end end do end # parse error
~~
until if true then 1.times { }   end do end # ok
until if true then 1.times do end end do end # parse error
~~
until until true do 1.times { }   end do end # ok
until until true do 1.times do end end do end # parse error
~~
until class Foo; 1.times { }   ; end do end # ok
until class Foo; 1.times do end; end do end # parse error
~~
until case; when true; 1.times { }   ; end do end # ok
until case; when true; 1.times do end; end do end # parse error
~~

This is because the underlined do's are not considered as block call but beginning of until' body.

Although this is confusing a little and can be actually fixed, the fix
needs many COND_PUSH(0)/COND_POP(), which may decrease performance and
code maintenability. In addition, writing such a long and complex
condition directly is absolutely bad (even insane) style.

So, I think there is no need to fix.

--
Yusuke Endoh mame@tsg.ne.jp
=end