when-after^{0,22}

when-after(e;*info*;*pred*?;*init*;*Trans*;*val*)
$\equiv_{\text{def}}$ if first(e)$\rightarrow$ $\langle \lambda x.\mathit{init}(\text{loc}(e),x),\ \mathit{Trans}(\text{loc}(e),\text{kind}(e),\mathit{val}(e),\lambda x.\mathit{init}(\text{loc}(e),x))\rangle$
  else let p = when-after(pred(e);*info*;*pred*?;*init*;*Trans*;*val*) in
    $\langle 2\text{of}(p),\ \mathit{Trans}(\text{loc}(e),\text{kind}(e),\mathit{val}(e),2\text{of}(p))\rangle$ fi
(recursive)

clarification:

when-after(e;*info*;*pred*?;*init*;*Trans*;*val*)
$\equiv_{\text{def}}$ if first(*pred*?;e)$\rightarrow$
    $\langle \lambda x.\mathit{init}(\text{loc}(\mathit{info};e),x)$
    , $\mathit{Trans}(\text{loc}(\mathit{info};e),\text{kind}(\mathit{info};e),\mathit{val}(e),\lambda x.\mathit{init}(\text{loc}(\mathit{info};e),x))\rangle$
  else let p = when-after(pred(*pred*?;e);*info*;*pred*?;*init*;*Trans*;*val*) in
    $\langle 2\text{of}(p),\ \mathit{Trans}(\text{loc}(\mathit{info};e),\text{kind}(\mathit{info};e),\mathit{val}(e),2\text{of}(p))\rangle$ fi
(recursive)

---

http://www.cs.cornell.edu/Info/Projects/NuPrl/FDLcontent/p0_286125_/p55_283063_{when-after}.html