We propose to define a new programming connective $P \diamond Q$. What properties of $\diamond$ are essential? Why?

It must be defined for all specifications $P$ and $Q$, not just for programs, so that it can be used during program development. It must be implementable, which means

$$(\forall \sigma \exists \sigma' \cdot P \land t' \geq t) \land (\forall \sigma \exists \sigma' \cdot Q \land t' \geq t) \Rightarrow (\forall \sigma \exists \sigma' \cdot (P \diamond Q) \land t' \geq t)$$

(This property can be contested because ensure is not implementable.) It must be monotonic in both operands so that Refinement by Steps and Refinement by Parts can be used.

If $A \Leftarrow B \diamond C$ and $B \Leftarrow D$ and $C \Leftarrow E$ are theorems,

then $A \Leftarrow D \diamond E$ is a theorem.

If $A \Leftarrow B \diamond C$ and $D \Leftarrow E \diamond F$ are theorems,

then $A \land D \Leftarrow B \land E \diamond C \land F$ is a theorem.

(Since $\diamond$ is a symmetric symbol, perhaps it ought to be a symmetric operator $P \diamond Q = Q \diamond P$ but that's not an essential point and there are lots of counterexamples.)