We propose to define a new programming connective $P \uparrow Q$. What properties of $\uparrow$ 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

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

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

If $A \iff B \uparrow C$ and $B \iff D$ and $C \iff E$ are theorems,
then $A \iff D \uparrow E$ is a theorem.

If $A \iff B \uparrow C$ and $D \iff E \uparrow F$ are theorems,
then $A \land D \iff B \land E \uparrow C \land F$ is a theorem.

(Since $\uparrow$ is a symmetric symbol, perhaps it ought to be a symmetric operator

$$P \uparrow Q = Q \uparrow P$$

but that's not an essential point and there are lots of counterexamples.)