(disjoint composition) Independent composition \( P \langle Q \rangle \) requires that \( P \) and \( Q \) have no variables in common, although each can make use of the initial values of the other's variables by making a private copy. An alternative, let's say disjoint composition, is to allow both \( P \) and \( Q \) to use all the variables with no restrictions, and then to choose disjoint sets of variables \( v \) and \( w \) and define
\[
P \langle v \rangle \langle w \rangle | Q = (P. \ v' = v) \land (Q. \ w' = w)
\]

(a) Describe how \( P \langle v \rangle \langle w \rangle | Q \) can be executed.

§ Make a copy of all variables. Execute \( P \) using the original set of variables and in parallel execute \( Q \) using the copies. Then copy back from the copy \( w \) to the original \( w \). Then throw away the copies. There may be variables \( x \) other than \( v \) and \( w \); if so, their final values are arbitrary, and this implementation makes them be what \( P \) says they should be. Formally, using application \( \langle v \rightarrow P \rangle \) as the formal notation for (substitute \( x \) for \( v \) in \( P \)),
\[
\begin{align*}
\text{var } &cw:= w' \text{ var } cv:= v' \text{ var } cx:= x' \\
&\quad (P \| \langle v, w, x, v', w', x' \rightarrow Q \rangle cv cx cv' cx').
\end{align*}
\]

(b) Prove that if \( P \) and \( Q \) are implementable specifications, then \( P \langle v \rangle \langle w \rangle | Q \) is implementable.

§ First, a lemma.
\[
P, \ v' = v \quad \text{expand dependent composition}
\]

\[
\begin{align*}
&= \exists v'', w'', x''. \langle v', w', x' \rightarrow P \rangle v'' w'' x'' \land v'' = v'' \\
&= \exists w'', x''. \langle v', w', x' \rightarrow P \rangle v' w' x'' \\
&= \exists w', x'. \langle v', w', x' \rightarrow P \rangle v' w' x' \\
&= \exists w', x'. P
\end{align*}
\]

So \( P \langle v \rangle \langle w \rangle | Q = (P. \ v' = v) \land (Q. \ w' = w) = (\exists w', x'. P) \land (\exists v', x'. Q) \)

Now the main proof.
\[
(P \langle v \rangle \langle w \rangle | Q \text{ is implementable}) \quad \text{definition of implementable}
\]

\[
\begin{align*}
&= \forall v, w, x. \exists v', w', x'. P \langle v \rangle \langle w \rangle | Q \\
&= \forall v, w, x. \exists v'. \exists w', x'. (\exists w', x'. P) \land (\exists v', x'. Q) \\
&= \forall v, w, x. \exists v'. \exists w'. \exists x'. (\exists w', x'. P) \land (\exists v', x'. Q) \\
&= \forall v, w, x. \exists v'. \exists w'. (\exists v', x'. Q) \land (\exists w', x'. P) \\
&= \forall v, w, x. (\exists v', x'. Q) \land (\exists w', x'. P) \\
&= \forall v, w, x. (\exists v', w, x'. P) \land (\exists v', x'. Q) \\
&= (\exists v, w, x. \exists v', w, x'. P) \land (\exists v, w, x. \exists v', x'. Q) \\
&= (P \text{ is implementable}) \land (Q \text{ is implementable})
\end{align*}
\]