Let $S$ be a set of states and $D(S)$ the set of probability distributions over $S$. The semantics of [8] assign to each program $P$ a function $S \rightarrow P(D(S))$. The semantics of sequential composition is given as

$$(P; Q)(s) = \left\{ \sum_{m \in S} f(m) \cdot g_m \mid f \in P(s) \land (\forall m : g_m \in Q(m)) \right\}$$  \hspace{1cm} (1)$$

We map a program $P$ to a relation $\rightarrow_P \subseteq D(S) \times D(S)$ as:

$$d \rightarrow_P \sum_{s \in S} d(s) \cdot d_s \quad \text{where} \quad \forall s : d_s \in P(s)$$

Now the question is:

$$d_1 \rightarrow_{P;Q} d_3 \quad ? \quad \exists d_2 : d_1 \rightarrow_P d_2 \rightarrow_Q d_3$$

Assuming $d_1 \rightarrow_{P;Q} d_3$, by definition

$$d_3 = \sum_{s \in S} d_1(s) \cdot d_s \quad \text{with} \quad d_s \in (P; Q)(s)$$

Note that $(P; Q)(s)$ contains all distributions produced by all possible choices of $f$ and $g_m$ in (1). For two different states $s, s'$, we freely select $d_s \in (P; Q)(s), d_{s'} \in (P; Q)(s')$, so the $g_m$’s used in the definition of $d_s$ and $d_{s'}$ can be different. Thus we have

$$d_3 = \sum_{s \in S} d_1(s) \sum_{m \in S} f_s(m) \cdot g_{s,m} \quad \text{with} \quad f_s \in P(s), g_{s,m} \in Q(m)$$  \hspace{1cm} (2)$$

where we write $g_{s,m}$ to emphasize that $g_{s,m}$ can depend on both $s$ and $m$, as explained above.

On the other hand, assuming $d_1 \rightarrow_P d_2 \rightarrow_Q d_3$, by definition

$$d_2 = \sum_{s \in S} d_1(s) \cdot f_s \quad \text{with} \quad f_s \in P(s)$$

$$d_3 = \sum_{m \in S} d_2(m) \cdot g_m \quad \text{with} \quad g_m \in Q(m)$$

$$= \sum_{m \in S} \sum_{s \in S} d_1(s) \cdot f_s(m) \cdot g_m$$

$$= \sum_{s \in S} d_1(s) \sum_{m \in S} f_s(m) \cdot g_m$$  \hspace{1cm} (3)$$

Comparing (2) and (3) we see that by taking $g_{s,m}$ to be independent of $s$ we can show that

$$d_1 \rightarrow_{P;Q} d_3 \iff \exists d_2 : d_1 \rightarrow_P d_2 \rightarrow_Q d_3$$

But the inverse does not hold in general.
**Convex-closed programs**  One extra property of programs created by
the semantics of [8] is that they are convex-closed:

\[ d_1 \in P(s) \land d_2 \in P(s) \Rightarrow xd_1 + (1 - x)d_2 \in P(s) \]

Now we can write (2) as

\[ d_3 = \sum_{m \in S} \sum_{s \in S} d_1(s) \cdot f_s(m) \cdot g_{s,m} \quad \text{with} \quad f_s \in P(s), g_{s,m} \in Q(m) \]

\[ = \sum_{m \in S} c_m \sum_{s \in S} f_s(m) \cdot \frac{d_1(s)}{c_m} \cdot g_{s,m} \quad \text{where} \quad c_m = \sum_{s \in S} d_1(s) \cdot f_s(m) \]

then \( \sum_{s \in S} \frac{f_s(m) \cdot d_1(s)}{c_m} \cdot g_{s,m} \) is a convex combination of elements of \( Q(m) \), so it is equal to some element \( h_m \in Q(m) \), thus

\[ d_3 = \sum_{m \in S} c_m \cdot h_m \quad h_m \in Q(m) \quad (4) \]

From (1):

\[ d_3 = \sum_{s \in S} d_1(s) \sum_{m \in S} f_s(m) \cdot g_m \]

\[ = \sum_{m \in S} g_m \sum_{s \in S} d_1(s) \cdot f_s(m) \]

\[ = \sum_{m \in S} c_m \cdot g_m \]

so by taking \( g_m \) equal to \( h_m \) (found in (4)) we conclude that

\[ d_1 \rightarrow_{P;Q} d_3 \quad \Rightarrow \quad \exists d_2 : d_1 \rightarrow_P d_2 \rightarrow_Q d_3 \]

Hence, in the case of convex-closed programs, \( \rightarrow_{P;Q} = \rightarrow_P \rightarrow_Q \).

**Non convex-closed**  If we drop this condition, consider a simple system
with states 0, 1, and programs \( P, Q \) with semantics:

\[ P(s) = \{ \delta(0) \} \]

\[ Q(s) = \{ \delta(0), \delta(1) \} \]

note that \( Q(s) \) is impossible to construct using the language of [8]. By defi-
nition:

\[ (P;Q)(s) = \{ \delta(0), \delta(1) \} \]

Letting \( d = \frac{1}{2}\delta(0) + \frac{1}{2}\delta(1) \) we have

\[ d \rightarrow_{P;Q} d \]

but the above does not hold for \( \rightarrow_P \rightarrow_Q \).