

# Review question 4

donderdag 9 februari 2023 13:01

$$\begin{aligned} & \text{new } n; (\text{out } c \ n; \mid \mid (\text{new } m; \text{out } c \ m;)) \\ \equiv & \text{new } n; (\text{out } c \ n; \mid \text{stop}) \\ \equiv & \text{new } n; \text{out } c \ n; \\ \equiv & \mid (\text{new } n; \text{out } c \ n;) \end{aligned}$$

these are equivalent

$$\begin{aligned} & \text{out } c \ m; \mid \text{new } n; \text{out } c \ n; \\ \equiv & \text{new } n; (\text{out } c \ m; \mid \text{out } c \ n;) \\ \neq & \text{new } n; (\text{out } c \ m; \mid \text{out } n \ c;) \end{aligned}$$

these are not equivalent

$$\begin{aligned} & \text{out } c \ m; \mid \text{new } n; \text{out } c \ m; \\ \equiv & \text{new } n; (\text{out } c \ m; \mid \text{out } c \ m;) \end{aligned}$$

these are equivalent

$$\text{out } c \ n; \mid \text{new } n; \text{out } c \ m; \neq \text{new } n; (\text{out } c \ n; \mid \text{out } c \ m;)$$

these are not equivalent; the first only binds  $n$  in the right process, so  $n$  is a free name in the left process, whereas the second one binds  $n$  in both parallelly composed processes.

$$\text{out } c \ n; \text{new } n; \text{out } c \ n; \neq \text{new } n; (\text{out } c \ n; \mid \text{out } c \ n;)$$

in the first instance of  $\text{out } c \ n;$   
 $\rightarrow n$  is free on the left side

$\rightarrow n$  is bound on the right side

$$\text{new } n; \text{new } m; (\text{out } c \ n; \mid \mid \text{out } c \ m;) \neq \mid (\text{new } n; \text{out } c \ n;)$$

in the left side, we only define  $n$  <sup>bind exactly</sup> once, while we do so zero or multiple times on the right side, in case of  $n$

$P_{1a}, P_{1b}, P_{1c}$  are not structurally congruent to each other