

Modeling Concurrent and Probabilistic Systems

Winter Term 07/08

– Series 8 –

Hand in until December 14 before the exercise class.

Exercise 1

(1+3 points)

Consider the following π -calculus process

$$P = \mathbf{new} z ((\bar{x} \langle y \rangle + z(w).\bar{w} \langle y \rangle) \parallel x(u).\bar{u} \langle v \rangle \parallel \bar{x} \langle z \rangle).$$

- a) Prove that $P \equiv x(u).\bar{u} \langle v \rangle \parallel \mathbf{new} z ((\bar{x} \langle y \rangle + z(w).\bar{w} \langle y \rangle) \parallel \bar{x} \langle z \rangle)$.
- b) Formally derive all successors of P using the reaction rules of the π -calculus.

Exercise 2

(2 points)

Prove that if x is not free in Q then $\mathbf{new} x Q \equiv Q$.

Exercise 3

(2 points)

By using structural congruence, exhibit the redex in

$$x(z).\bar{y} \langle z \rangle \parallel !(\mathbf{new} y (\bar{x} \langle y \rangle .Q))$$

and formally derive the result of the reaction!

Exercise 4

(4 points)

We wish to send messages consisting of more than one name. So we want to allow the forms

$$x(y_1, \dots, y_n).P \text{ and } \bar{x} \langle z_1, \dots, z_n \rangle .Q$$

(where all the y_i are distinct) for any $n \geq 0$. For a correct encoding, we have to ensure that there cannot be an inference on the channel along which a composite message is sent. To send a message $\langle z_1, \dots, z_n \rangle$, we first send a *fresh* name w along x , then send the components z_i one by one along w . So we translate the multiple action prefixes as follows:

$$\begin{aligned} x(y_1, \dots, y_n).P &\mapsto x(w).w(y_1) \dots w(y_n).P \\ \bar{x} \langle z_1, \dots, z_n \rangle .Q &\mapsto \mathbf{new} w (\bar{x} \langle w \rangle .\bar{w} \langle z_1 \rangle \dots \bar{w} \langle z_n \rangle .Q) \end{aligned} \quad \text{where } w \notin \mathbf{fn}(Q)$$

Apply this encoding to

$$x(y_1 y_2).P \parallel \bar{x} \langle z_1 z_2 \rangle .Q \parallel \bar{x} \langle z'_1 z'_2 \rangle .Q'$$

Do at least two reduction sequences to convince yourself that only the right replacements occur!