Ninth Homework of Concurrent Systems

Exercise 1 Consider the classical implementation of booleans in the polyadic π-calculus:

$$\begin{array}{rcl} True_a & \triangleq & a(y,z).\overline{y}\langle\rangle \\ False_a & \triangleq & a(y,z).\overline{z}\langle\rangle \\ Test_a(P;Q) & \triangleq & (\nu b,c)\overline{a}\langle b,c\rangle.(b().P \mid c().Q) \quad \text{for } \{b,c\} \cap fn(P,Q) = \emptyset \end{array}$$

First, provide the full inference for deriving the first reduction of process $Test_a(P;Q) \mid True_a$; then, draw the complete LTS of process $Test_a(\mathbf{0};\mathbf{0}) \mid True_a \mid False_a$, by assuming in the LTS the expectable rules for polyadic inputs and outputs.

Exercise 2 Give a type environment for typing the polyadic π-calculus process $a(x,y).(\nu b)(\overline{x}\langle b\rangle \mid !\,y(u).\overline{b}\langle u\rangle)$. Then provide a type inference for its typeability.

Exercise 3 Formally define the free names (written $fn(\cdot)$) of a π-calculus process P by induction on the process syntax (essentially, these are those not bound by a restriction or by an input). Then show that, if $P \xrightarrow{\alpha} P'$, then it may be possible that $fn(P') \not\subseteq fn(P)$.

Exercise 4 Formally define the run-time error for processes of the higher-order π-calculus (HINT: take inspiration by the run-time errors of the polyadic calculus).