Process Algebras and Concurrent Systems
An exercise for the GLOBAN participants
by R. DE NICOLA

Consider the following two languages whose syntax is defined below and whose semantics is reported in the handouts.

micro-CCS

$$P ::= nil \mid a.P \mid \tau.P \mid P_1 + P_2 \mid P_1 | P_2 \mid P\backslash K \mid P[f]$$

micro-CSP

$$Q ::= nil \mid a.Q \mid \sum_{i \in I} Q_i \mid Q_1 \Box Q_2 \mid Q_1 \| Q_2 \mid Q/A \mid Q[f]$$

1. Define a translation function "tr" mapping one of the languages into the other (you are free to choose source and target language)

2. Study correctness of the proposed translation by checking whether there exists a behavioural equivalence "$\simeq$" (weak/strong bisim., testing, traces, ...) such that you have

$$P_1 \simeq P_2 \text{ iff } tr(P_1) \simeq tr(P_2)$$

N.B. If useful (necessary) you can consider "feasible" versions or sublanguages of micro-CCS or micro-CSP, e.g. by requiring that summands have given terms have different initial actions.