Prof. Dr. Ir. Joost-Pieter Katoen
Benjamin Kaminski, Hao Wu

Theoretical Foundations of the UML SS 2016
— Series 3 —

Hand in until May 27 before the exercise class.

Exercise 1 (Definedness of CMSC–Concatenation) (2+2 Points)

1. Prove or disprove: There exists a CMSC M_1 with process set $\mathcal{P}_1 = \{p_1, p_2\}$, such that for all CMSC M_2 which satisfy the following side conditions, it holds that $M_1 \bullet M_2$ is not defined. The side conditions are:
   - For the process set $\mathcal{P}_2$ of M_2 it holds that $\mathcal{P}_2 = \mathcal{P}_1$, and
   - M_2 contains an unmatched receive event of the form "p_2 receives message content a from p_1".
2. Prove or disprove: There exists a CMSC M_1 with process set $\mathcal{P}_1 = \{p_1, p_2\}$, such that for all CMSC M_2 which satisfy the same side conditions as given above, it holds that $M_2 \bullet M_1$ is not defined.

Exercise 2 (Non–Associativity of CMSC–Concatenation) (2 Points)

Prove or disprove: There exist three CMSCs M_1, M_2, and M_3, such that

$$(M_1 \bullet M_2) \bullet M_3 \neq M_1 \bullet (M_2 \bullet M_3) ,$$

even though both $(M_1 \bullet M_2) \bullet M_3$ and $M_1 \bullet (M_2 \bullet M_3)$ are defined.

Exercise 3 (Safe Path Existence in CMSGs) (2+3 Points)

1. Give a *formal* description of the v_n's used in the proof of undecidability of the existence–of–a–safe–path–problem.
2. Prove or disprove: The decision problem "Does CMSG G have at least *two* safe, accepting paths?" is undecidable.

   Hint: For proving decidability, it suffices to give an informal but precise description of an algorithm deciding this problem. You *do not* have to provide an algorithm in an actual programming language. For proving undecidability, it suffices to choose an appropriate undecidable problem and give a correct reduction function and an informal but precise argument why your reduction function is correct. You *do not* have to formally prove the correctness of your reduction.