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

Concurrency Theory WS 2013/2014

— Series 2 —

Hand in until November 7th before the exercise class.

Exercise 1 (CCS and LTS) (2+3 Points)

1) Consider following CCS process definitions induce infinite LTS or not? Justify your answers.

a) $B(a) = (B(a) \parallel B(a)) + a.\text{nil}$

b) $D(a, b) = a.(D(a, b) \parallel b.\text{nil})$

2) Show that it is undecidable whether a given CCS process definition induces a finite or an infinite LTS. (*Hint*: a tape of a Turing machine can be modelled as two stack processes)

Exercise 2 (Modelling and verification) (3+2 Points)

One of the mutual exclusion algorithms is called Hyman's algorithm which was proposed in 1966. In Hyman's algorithm each P_i ($i \in \{1, 2\}$) executes the following algorithm (j denotes the index of other process, and the initial value of k is immaterial)

```
while true do
begin
       <noncritical section>;
       b_i := true;
       while k ≠ i do begin
                        while b_j do skip;
                        k := i;
                      end;
       <critical section>;
       b_i := false;
end;
```

1) Give a CCS definition to describe the behavior of Hyman's "mutual exclusion" algorithm.

2) Give HML formula to express the property of mutual exclusion that the algorithm should satisfy. Argue that whether the Hyman's algorithm satisfies this property.