



ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Prof. Dr. Andreas Podelski
Matthias Heizmann
Christian Schilling

July 8th-9th, 2014

6. Presence Exercise Sheet for the Lecture Computer Science Theory

Exercise 1: Halting Problem for Programs

We consider the alphabet $B = \{0, 1\}$ and programs written in an imperative programming language IMP.

A IMP *program code* is a binary string according to the rules for proper IMP program code (these rules are of no concern for us). It is decidable whether a given binary string is a proper IMP program code or not (by a compiler).

A IMP *program* is the usual interpretation of IMP program code according to the semantic rules of IMP programs (these rules are of no concern for us).

Every IMP program has some IMP program code associated. Also, every IMP program code encodes some IMP program.

To distinguish between programs and their code, we use variable names (e.g., P) for programs and the same variable names with a line above (e.g., \overline{P}) for program code.

IMP programs can be executed. They can read an (binary) input word and can produce an (binary) output word. Thus, IMP programs can also read their own encoding as input.

Show that there is no IMP program P_{magic} that executes the following algorithm:

Input: a IMP program code $\overline{P} \in B^*$

Output: 0 or 1, depending on the question: Does P terminate with the input \overline{P} ?

Hint: Adapt the proof for the special halting problem.