論文の内容の要旨

論文題目　Program Analysis by Size-Change Termination
（サイズ変化停止性によるプログラム解析）

氏名　フレデリクセン、カール　クリスチャン

The purpose of this thesis is to apply the Size-Change Termination principle to implement various program analyses more precisely and more efficiently. For this purpose, we extend the existing framework of the principle by identifying sub-computations in the subject program. We focus in particular on polynomial running time analysis and verification of liveness properties. Analysis of sub-computations can make polynomial running time analysis more precise and liveness verification both more efficient and precise.

Size-Change Termination is a principle for approximating program termination, originally formulated for first order functional programs with well-founded data -- a program can be shown to terminate if every infinite computation is infeasible. First, a set of size relations, Size-Change Graphs, between parameters and the arguments passed in function calls is extracted. By defining a composition operator on size-change graphs any finite sequence of calls can be approximated by the graph resulting from composing the graphs corresponding to the calls in the sequence. The central, and non-trivial, theorem of size- change termination then states that by computing the *closure set* of the size-change graphs, any infinite computation can be approximated by a single size-change graph in the closure set. We can then test for termination by checking if certain graphs in the closure set cause a variable to strictly decrease in size, thus violating the well-foundedness assumption.

The purpose of the polynomial running time analysis is to establish whether a given program will terminate in polynomially many computational steps. One important work is a syntactical restriction by Bellantoni-Cook which can express precisely all PTIME *algorithms*, but fails to recognize many classical polynomial time *programs*. We propose a program analysis that can recognize many polynomial time programs, including all Bellantoni-Cook program and programs with non-linear recursion. The analysis consists of 3 tests: First we apply Size-Change Termination to test if the call-depth is polynomially bounded. Second, we ensure that data values that control the recursion cannot be shared in non-linear recursive calls. Finally, we place restrictions on how variables that can "grow too fast" can be passed to sub-computations.

For verification of liveness properties the purpose is to decide whether *undesired* infinite computations are possible for a given program, where “undesired” encodes the liveness property of interest. Based on Size-Change Termination, Podelski have developed a method for verification of liveness properties. The central idea is to abstract changes in size over program transitions, use Size-Change Termination to rule out infeasible infinite computations and finally test the liveness property against the remaining infinite computations. We propose a method for improving on efficiency and precision by identifying *sub-computations* which can be analyzed individually of the context in which they occur. Precision can then be increased by applying linear programming to obtain summary information for the effect of sub-computations. We also propose an extension of the sub-computation based verification algorithm to parallel programs communicating via named channels, which analyzes interleavings of sub-computations from different processes that communicate via the same channel.