

Ladění velkých programů je časově náročný úkol, který dosud nebyl plně automatizován. Vývojář musí nejprve reprodukovat danou běhovou chybu. Poté musí najít hlavní příčinu chyby a správně ji opravit. Automatizace může tento proces výrazně zefektivnit snížením množství kódu, se kterým musí vývojář pracovat. Tato práce představuje tři metodiky automatického redukování daného selhávajícího programu na jeho minimální spustitelnou podmnožinu. Techniky jsou založeny na existujících poznatcích v oblasti ladění. Automaticky minimalizovaný program musí vést ke stejné běhové chybě jako původní program. Minimalizace se zaměřuje na optimální výsledky v oblasti malých a jednoduchých aplikací.

Cílem této práce je popsat techniky, které jsou vhodné pro redukcí programu. Tyto techniky jsou poté implementovány pomocí Clang LibTooling, knihovny pro nástroje postavené na projektu Clang. Práce vysvětluje vnitřní fungování každé implementace a poukazuje jejich omezení. Implementace jsou porovnány na sadě zdrojových souborů psaných v jazycích C a C++. Efektivita implementace je odvozena na základě velikosti generovaného výstupu a doby chodu algoritmu.