

Moderní překladače aplikují při generování kódu sadu optimalizací, jejichž cílem je zrychlit výsledný program. Kombinovaný účinek jednotlivých optimalizací je však často nepředvídatelný, změny kódu kompilátoru tak mohou jako neúmyslný důsledek zpomalit výsledný program. Vzhledem k velkému počtu kompilačních jednotek a aplikovaných optimalizací je obtížné takové regrese diagnostikovat.

Tato práce pomáhá v diagnostice výkonnostních regresí zachycením optimalizačních rozhodnutí kompilátoru. Činíme tak reprezentací použitých optimalizačních fází, optimalizačních rozhodnutí a rozhodnutí o inlinování ve formě stromů, a aplikováním stromové editační vzdálenosti (TED) na tyto stromy k poloautomatickému odhalování optimalizačních rozdílů. Protože tentýž zdrojový kód může být inlinován v různých kontextech a tedy i různě optimalizován, představujeme rovněž způsob porovnávání optimalizací téhož kódu v různých kontextech. Uvedené techniky používáme k určení příčin výkonnostních problémů v benchmarcích překladače Graal.