

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Filip Kastl
Název práce Alternativní algoritmus stavby SSA formy pro GCC
Rok odevzdání 2023
Studijní program Informatika
Specializace Obecná informatika

Autor posudku Jan Hubička Vedoucí
Pracoviště Katedra aplikované matematiky

K celé práci

lepší OK horší nevyhovuje

| | lepší | OK | horší | nevyhovuje |
|--|-------|----|-------|------------|
| Obtížnost zadání | X | | | |
| Splnění zadání | X | | | |
| Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i> | X | | | |
| <p>GNU Compiler Collection (GCC) je produkční překladač s více než třicetiletou historií. Jakákoliv netriviální práce na takovém projektu vyžaduje důkladné studium zdrojových kódů, formátovacích konvencí a seznámení se s postupy práce na takovém projektu.</p> <p>Single Static Assignment (SSA) je speciální forma mezyazyka, která vyžaduje, že každá proměnná je definovaná právě jednou. SSA je dnes standardní reprezentací, kterou využívají téměř všechny moderní překladače. Do GCC byla zabudována v letech 2001–2004 a jednalo se o největší změnu v historii GCC. Implementace nového algoritmu zasahuje do naprosto klíčové části překladače a je náročná zejména nutností pochopit a správně implementovat jednotlivé detaily jazyka GIMPLE. Vzhledem k této skutečnosti zadání nepočítá s úplným nahrazením původního algoritmu, protože takový projekt zásadně přesahuje rozsah bakalářské práce. Cílem práce bylo vytvoření alternativního API pro generování mezikódu a otestování algoritmu v praxi. I v tomto rozsahu se jedná o náročný úkol.</p> | | | | |

Textová část práce

lepší OK horší nevyhovuje

| | lepší | OK | horší | nevyhovuje |
|---|-------|----|-------|------------|
| Formální úprava <i>... jazyková úroveň, typografická úroveň, citace</i> | X | | | |
| Struktura textu <i>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i> | X | | | |
| Analýza | X | | | |
| Vývojová dokumentace | X | | | |
| Uživatelská dokumentace | X | | | |

GCC implementuje standardní algoritmus pro konstrukci SSA formy podle článku **Cytron–Ferrante–Rosen–Wegman–Zadeck: Efficiently computing static single assignment form and the control dependence graph**. Tento článek chytře využívá velmi netriviální algoritmus Lengauera a Taraja pro stavbu dominátorů ve skoro-lineárním čase. Díky tomu ale vyžaduje znalost celého těla funkce.

Nové algoritmy, motivované zejména JIT (just-in-time) překladači, se zaměřily na lepší integraci konstrukce SSA do procesu generování mezikódu. Článek **Braun–Buchwald–Hack–Leiße–Mallon–Zwinkau: Simple and Efficient Construction of Static Single Assignment Form** nabízí zajímavou a elegantní alternativu, která jako první slibuje tvorbu minimální SSA formy a integraci základních optimalizací. Výsledek takového algoritmu slibuje rychlejší běh překladače a menší spotřebu paměti.

Filip Kastl implementoval algoritmus jako dvě oddělené části: první přináší nové API pro generování mezikódu, které integruje jednotlivé optimalizace. Druhá část pak implementuje optimalizaci silně souvislých komponent kopií, které běžně vznikají během překladače a způsobují, že SSA forma není minimální.

Textová část práce čitelným způsobem diskutuje oba algoritmy a popisuje změny nutné pro jejich nasazení v GCC a pro generování kódu do již existující funkce v SSA formě. Uživatelská část popisuje API pro generování mezikódu včetně krátkého tutoriálu.

Implementační část práce

lepší OK horší nevyhovuje

| | | | | |
|---|---|---|--|--|
| Kvalita návrhu ... <i>architektura, struktury a algoritmy, použité technologie</i> | X | | | |
| Kvalita zpracování ... <i>jmenné konvence, formátování, komentáře, testování</i> | X | | | |
| Stabilita implementace | | X | | |

API pro generování mezikódu sice není úplné, ale je dostatečně silné aby nahradilo generování kódu v některých částech překladače a dokládá praktičnost implementovaného algoritmu. Překvapením bylo také množství vyoptimalizovaných kopií, které výrazně předčilo původní očekávání.

Praktické nasazení algoritmu je teď aktivně diskutováno komunitou vývojářů GCC a lze předpokládat že bude přijato do oficiálního vydání ve verzi 15.1.

Filip Kastl pracoval intenzivně již od prvního semestru softwarového projektu a prokázal schopnost zorientovat se v komplikovaném softwarovém projektu i schopnost návrhu kvalitního API a implementace netriviálního algoritmu.

Celkové hodnocení Výborně

Práci navrhuji na zvláštní ocenění Ne

Datum

Podpis