

Optimierungen in Übersetzern: Verfahren

Marco Ammon (my04mivo)

10. August 2020

Inhaltsverzeichnis

1	Kontrollflussanalyse	2
1.1	Kontrollflussgraph	2
1.2	Dominanz	2
1.3	Schleifenerkennung	2
1.4	Kontrollflussabhängigkeit	3
2	Datenflussanalyse	3
2.1	Datenabhängigkeiten	3
3	Aliasanalyse	3
4	Induktionsvarianten und schleifeninvarianter Code	3
5	Schleifen und Arrays	3
6	Schleifentransformationen	3
7	Schleifenrestrukturierungen	3

1 Kontrollflussanalyse

1.1 Kontrollflussgraph

- Gerichteter Graph
- Knoten: Grundblöcke (meist maximal)
- Kante zwischen zwei Blöcken A und B wenn B direkt nach A ausgeführt werden kann (etwa [un-]bedingter Sprung oder Fallthrough)
- Synthetische Ergänzung um Entry- und Exit-Knoten, die mit Kante verbunden sind

1.2 Dominanz

- Knoten x dominiert y ($x \geq y$), wenn jeder Pfad von Wurzel zu y durch x laufen muss
- Strikte Dominanz $x \gg y$, falls zusätzlich $x \neq y$ gilt
- $\text{ImmDom}[y]$ ist strikter Dominator von y , der y am Nächsten ist
- Bestimmung des ImmDomms: TODO
- Dominatorbaum enthält jeden Knoten als Kind seines ImmDomms \rightarrow Pfad zwischen x und z in Dominatorbaum $\Leftrightarrow x \gg z$

1.3 Schleifenerkennung

- Region:
 - Untergraph mit einem „Header“ d , der (potentiell mehrere) Eingangskante von außerhalb besitzt
 - Wichtige Region: maximale Region mit d dominiert alle Knoten der Region
 - Hierarchischer Flussbaum: Baum der Regionen
- Rückwärtskante: Kante (n, d) mit $d \geq n$
- Natürliche Schleife:
 - Rückwärtskante (n, d) sowie alle Knoten k mit $d \geq k$ und es gibt einen Pfad von k nach n ohne d
 - Bestimmung mit Worklist-Algorithmus, der bei n beginnt und rekursiv die Vorgänger bis d durchläuft und in Menge aufnimmt
- Suche nach Rückwärtskanten und natürlichen Schleifen in wichtigen Regionen ausreichend
- „Unsaubere“ Regionen:
 - ein Knoten dominiert nachgeordneten Zyklus
 - Erkennung durch Prüfung der Reduzierbarkeit des Graphs:
 - * Entfernung der Rückwärtskanten aus KFG \rightarrow azyklischer Graph, in dem jeder Knoten von der Wurzel erreicht werden kann \Leftrightarrow KFG frei von unnatürlichen Schleifen
 - * Alternative mit Transformationen: Am Ende Graph aus einem einzigen Knoten \Leftrightarrow KFG reduzierbar (ohne Zyklen)
 - T1-Transformation** Selbstschleifen aus Graph löschen
 - T2-Transformation** Knoten mit eindeutigem Vorgänger mit diesem zusammenfassen

1.4 Kontrollflussabhängigkeit

2 Datenflussanalyse

2.1 Datenabhängigkeiten

- Schreiben vor Lesen:
- Schreiben vor Schreiben: Ausgabeabhängigkeit
- Lesen vor Schreiben: Anti-Abhängigkeit

3 Aliasanalyse

4 Induktionsvarianten und schleifeninvarianter Code

5 Schleifen und Arrays

6 Schleifentransformationen

7 Schleifenrestrukturierungen