

Prüfungsfragen Algorithmische Sprachen 2005

Übersetzerbau, Ausgewählte Kapitel aus dem Übersetzerbau
Prof. Dr. Michael Philippsen, Beisitzer: Dipl.-Inf. Michael Klemm
Januar 2005

Bemerkungen zu Prüfung und Prüfer

- Schwerpunktfach
- Ergebnis: 1,3
- angenehme Prüfungsatmosphäre
- fragt viel Verständnis, auch mal Dinge die nicht direkt im Skript stehen, gibt dann aber auch Hilfestellungen

Fragen

- Worauf muss man achten, wenn man Code parallelisieren will? → Datenabhängigkeiten (Fluss-, Ausgabe-, Anti-)
- Wie erkennt man Datenabhängigkeiten auf einer Kellermaschine? Wann ist das Umordnen von push- und pop-Operationen legal? → Tüftelfrage, musste mir was überlegen.
- Statischen Vorgängerverweis erklären. Wieso ist diese Methode langsam? → Zeigerverfolgung, Cache wird „verschmutzt“.
- Bessere Methode? → Display
- Sichtbarkeit von Variablen → Aufbau einer Definitionstabelle, als Stack organisiert, lineare Suche nach Variablen langsam
- Graham-Glanville-Verfahren erklären; Vor- und Nachteile
- (ab hier Klemm) Wie berechnet man lebendige Variablen? → Datenflussanalyse, von unten nach oben Information propagieren, z. B. Bitvektor verodern.
- Wann werden die Bits gesetzt bzw. gelöscht? → Setzen bei Benutzung, Löschen bei Definition

- Was sind die Nachteile von Bitvektoren? → Viel Information wird herumgeschleppt, die nur an bestimmten Stellen benötigt wird, bei Umordnung des Codes müssen die Vektoren neu berechnet werden, viel Aufwand
- Besseres Verfahren? → SSA-Verfahren kurz erklärt, Vorteile z. B. beim Erkennen lebendiger Variablen

Compilerbau 1+2

Prof. Philippsen
Beisitzer: Dipl. Inf. Dominic Schell
Oktober 2005

Bemerkungen zu Prüfung und Prüfer

- Schwerpunktfach
- Ergebnis: 1,7
- angenehme Atmosphäre; hat nur noch ein Tässchen Kaffee gefehlt.
- eine indirekte Frage, der Rest war direkt

Fragen

- Wieder mal die Frage, ob irgendetwas ausgelassen werden sollte (Joker).
- Schleifenparallelisierung anhand eines Beispiels; Restrukturierungen, Vertauschung möglich?
- Graham Glanville: Nachteile, Schwierigkeiten, Probleme, Maschinengrammatik.
- Besucher Muster: Wieso, Weshalb, Warum.
- Welche Optimierungen zielen auf die Minimierung der Codegröße?
In Gedanken alle Optimierungen durchgehen. Der Prüfer ließ mit sich reden, was nun genau die Codegröße optimiert und was nicht.
- Warum kann das Vorziehen von gemeinsamen Teilausdrücken den Code vergrößern?
Diese Frage hat der Prüfer als sehr schwierig beurteilt, deswegen hier die Antwort:
Weil durch das Vorziehen der Registerdruck an der neuen Berechnungsstelle erhöht wird. Das

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen. Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du auf unserer Homepage.
--

hat zur Folge, dass die Registerallokation eventuell tonnenweise Spillcode erzeugen muss.

- Uebersetzung der case Anweisung.
Hier wollte der Prüfer unbedingt den fall-through hören.
- Registerallokation: Graphfärben

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen.
Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du auf unserer Homepage.