



“Optimierende Compiler”
Aufgabe 1: Triangle
Abgabe bis zum 03.05.2006, 12:00 MET mittags

In Ihrer Dreiergruppe sollen drei nicht-triviale Beispielprogramme in der Programmiersprache Triangle erstellt und mit dem Beispiel-Compiler auf der virtuellen Maschine TAM erprobt werden.

1 Einleitung

Für den Test der Funktionsfähigkeit eines Compilers im allgemeinen sowie der Leistungsfähigkeit verschiedener Optimierungsverfahren im speziellen wird in der Regel auf eine Sammlung verschiedenartiger Programmtexte zurückgegriffen.

In der bei uns verwendete Programmiersprache Triangle existiert ein solcher Fundus noch nicht. Zweck dieser Übung ist es also, neben Erfahrung mit Triangle zu sammeln, auch eine Test-Suite an Beispielprogrammen zusammenzustellen. Diese werden dann allen Teilnehmern zugänglich gemacht.

2 Problemstellung

Jedes Ihrer Programme soll einen Mindestumfang von ca. 160 Zeilen haben. Um den Test späterer Optimierungsverfahren zu evaluieren, sollten die Programme tatsächlich nicht-triviale Berechnungen durchführen, die in der Regel in Schleifen ablaufen.

Einige mögliche Problembereiche sind z.B. Datenkompression, Kryptographie, Mandelbrotmengen, Partitionierung via Min-Cut, das 8-Damen-Problem, verschiedene Graph-Algorithmen etc. Ihrer Phantasie sind fast keine Grenzen gesetzt.

Beachten Sie, dass weder Triangle noch TAM Dateioperationen bereitstellen. Sie können also lediglich durch Umlenken der Standardin- und -ausgabeströme Dateien in die Abarbeitung mit einbeziehen. Auch werden keine Gleitkommazahlen unterstützt. In einigen Fällen können Sie diese mittels geeignete Fixpunktdarstellungen durch Integer approximieren. Sie können natürlich auch eine Bibliothek implementieren, die einen eigenen Gleitpunkttyp samt dazugehöriger Arithmetik bereitstellt.

Beachten Sie, dass es nicht so sehr auf gute Benutzbarkeit Ihres Programmes ankommt. Beispielsweise kann der Schlüssel eines Kryptosystems durchaus als Klartext im Programm als Konstante definiert werden. Es handelt sich bei diesen Anwendungen in erster Linie um Testprogramme für den Compiler.

3 Abgabe

Finden Sie sich bitte zu Dreiergruppen zusammen. Falls es dabei Schwierigkeiten geben sollte, sprechen Sie den Dozenten nach der Vorlesung an.

Jede Dreiergruppe schickt spätestens zum Abgabezeitpunkt alle Ihre Dateien zusammen als ein `.jar` Archiv an

`oc06@esa.informatik.tu-darmstadt.de`

Jedes der Programme besteht aus ...

- Seinen `.tri` Triangle-Quellen
- Der `.tam` Objektdatei
- Einer README Textdatei, die beschreibt ...
 - Den Autor des Programmes
 - Die Funktion des Programmes
 - Die Bedienung des Programmes
 - Beispieleingaben und die erwarteten Ausgaben
- Optional noch Eingabe- und die dazu passenden Ausgabedateien

4 Beurteilung

Die Beurteilung dieser ersten Abgabe erfolgt via E-Mail. Kolloquien oder Vorträge finden noch nicht statt.