ulm university universität
uulm

Prof. Dr. Heiko Falk | Universität Ulm | 89069 Ulm

Fakultät für Ingenieurwissenschaften und Informatik
Institut für Eingebettete Systeme / Echtzeitsysteme

Prof. Dr. Heiko Falk

James-Franck-Ring
D-89081 Ulm

Tel.: +49 731 50-24183
Fax: +49 731 50-24182
Heiko.Falk@uni-ulm.de
http://www.uni-ulm.de/in/esys

Ulm, den 10. Mai 2012

Compiler für Eingebettete Systeme – Übungsblatt Nr. 1

Es soll eine Quellcode-Transformation realisiert werden, die mit Hilfe von Polytop-Analysen stets erfüllbare bzw. stets unerfüllbare Bedingungen von If-Statements entdeckt und durch Konstanten ersetzt *(„Condition Satisfiability")*. Hierzu wird das Grundgerüst dieser Prepass-Optimierung in Form von C++-Code bereitgestellt. Dieses Grundgerüst ist in den folgenden Aufgaben zu ergänzen, so dass sukzessive eine funktionierende Optimierung entsteht.

1. **Aufgabe: (Traversieren von ICD-C)** *(10 Punkte)*
   Um möglicherweise redundante Bedingungen in *If-Statements* zu entdecken, ist es notwendig, die High-Level IR ICD-C rekursiv zu traversieren. Realisieren Sie das Traversieren durch *Compound-*, *Loop-*, *If-Else-*, *If-* und *Switch*-Statements von ICD-C.
   Hinweis: Bearbeiten Sie zum Lösen dieser Aufgabe ausschließlich die Methoden
   - `IR_OptimizeIfStatements::iterateCompoundStmt( IR_CompoundStmt *, IR_CompoundStmt * )`
   - `IR_OptimizeIfStatements::iterateLoopStmt( IR_CompoundStmt *, IR_CompoundStmt * )`
   - `IR_OptimizeIfStatements::iterateIfElseStmt( IR_CompoundStmt *, IR_CompoundStmt * )`
   - `IR_OptimizeIfStatements::iterateIfStmt( IR_CompoundStmt *, IR_CompoundStmt * )`
   - `IR_OptimizeIfStatements::iterateSwitchStmt( IR_CompoundStmt *, IR_CompoundStmt * )`

   aus Datei `iroptimizeifstatements.cc`. Folgen Sie dabei den in obigen Methoden zu findenden Kommentaren und Hinweisen.
2. **Aufgabe: (Analysieren affiner Schleifen)** *(10 Punkte)*
   „Condition Satisfiability" ist nur unter der Voraussetzung anwendbar, dass die Grenzen aller Schleifen, die eine Bedingung umgeben, affin sind. Realisieren Sie das Überprüfen, ob untere bzw. obere Grenzen von Schleifen in ICD-C affin sind.
   Hinweis: Kommentieren Sie zunächst das Makro `DEBUG_ITERATION` am Beginn von Datei `iroptimizeifstatements.cc` aus und das Makro `DEBUG_ANALYSIS` ein. Bearbeiten Sie zum Lösen dieser Aufgabe ausschließlich die Methoden
   - `IR_OptimizeIfStatements::isAffineBound( IR_Exp *, IR_Symbol *, bool )`
   - `IR_OptimizeIfStatements::isSumOfIvarProds( IR_Exp *, IR_Symbol * )`

   aus Datei `iroptimizeifstatements.cc`. Folgen Sie dabei den in obigen Methoden zu findenden Kommentaren und Hinweisen.
3. **Aufgabe: (Polytop- und Code-Modifikation)** *(10 Punkte)*
   Zur Polytop-Analyse von Schleifen und Bedingungen müssen affine Schleifengrenzen und

Bedingungen von *If-Statements* als Ungleichungen für Polytope formuliert werden.

(a) Realisieren Sie die Erzeugung einzelner Polytop-Ungleichungen für einzelne affine Ausdrücke in ICD-C. *(8 Punkte)*
Hinweis: Kommentieren Sie zunächst das Makro `DEBUG_ANALYSIS` am Beginn von Datei `iroptimizeifstatements.cc` aus und das Makro `DEBUG_TRANSFORMATION` aus Dateien `iroptimizeifstatements.cc` und `constraint.cc` ein. Bearbeiten Sie zum Lösen dieser Teilaufgabe ausschließlich die Methoden

- `Constraint::insertFactors( IR_Exp *, IR_OptimizeIfStatements *, bool inverted )`
- `Constraint::normalize()`

aus Datei `constraint.cc`. Folgen Sie dabei den in obigen Methoden zu findenden Kommentaren und Hinweisen.

(b) Realisieren Sie das Ersetzen redundanter Bedingungen durch die Konstanten `0` bzw. `1` in ICD-C. *(2 Punkte)*
Hinweis: Bearbeiten Sie zum Lösen dieser Teilaufgabe ausschließlich die Methode

- `IR_OptimizeIfStatements::optimizeCondition( IR_Exp *, bool )`

aus Datei `iroptimizeifstatements.cc`. Folgen Sie dabei den in obiger Methode zu findenden Kommentaren und Hinweisen.

Organisatorische Hinweise:

- Die Übungen zu „Compiler für Eingebettete Systeme“ finden in Form von Blockübungen an folgenden Tagen und Uhrzeiten statt:
  Montag 21. Mai 2012 Montag 25. Juni 2012 Montag 16. Juli 2012
  jeweils von 13.00 - 16.30 Uhr.
- Die Übungen sind praktischer Natur und werden daher an den Rechnern im Raum 311 des Gebäudes O27 abgehalten. Pro Rechner bearbeitet ein Zweier-Team von Studierenden gemeinsam das jeweils aktuelle Übungsblatt.
  Das Bearbeiten der Übungsblätter setzt C++-Kenntnisse voraus.
- Pro Übungstermin ist ein Übungsblatt zu bearbeiten. Einen unbenoteten Schein, der die erfolgreiche Teilnahme an den Übungen zu „Compiler für Eingebettete Systeme“ bestätigt, kann erhalten, wer mindestens 50 Prozent der auf allen Übungszetteln insgesamt erreichbaren Punkte erzielt hat. Pro Übungsblatt sind maximal 30 Punkte zu erreichen.
- Rufen Sie nach jeder einzelnen Aufgabe Ihren Betreuer und demonstrieren Sie, wie sich Ihre Optimierung jeweils verhält.

Technische Hinweise:

- Loggen Sie sich mit den bereitgestellten Zugangsdaten an Ihrem Rechner ein. Loggen Sie sich von dort aus per SSH auf der folgenden Maschine ein:
  `wccsrv02.informatik.uni-ulm.de`
- Wechseln Sie ins Verzeichnis `WCC`; rufen Sie zum Bearbeiten der C++-Dateien der Optimierung *„Condition Satisfiability“* den Editor `kate` auf. Bearbeiten Sie die Aufgaben dieses Blattes genau in obiger Reihenfolge, da die Aufgaben aufeinander aufbauen.
- Um Ihre bearbeiteten C++-Dateien zu übersetzen, rufen Sie im Verzeichnis `WCC` das Programm `make` auf.
- Zum Überprüfen, wie sich Ihre Optimierung verhält, und ob sie die beabsichtigten Effekte zeigt, geben Sie im Verzeichnis `WCC` die folgende Kommandozeile ein: `wcc me.c`
  Der Ablauf Ihrer Optimierung wird durch Debug-Ausgaben im aktuellen Fenster dargestellt. Zusätzlich wird die Datei `test.ir` erzeugt, die das C-Programm `me.c` nach Ihrer Optimierung enthält.
- Zum Bearbeiten dieses Übungszettels benötigen Sie Hintergrundinformation zur verwendeten High-Level IR ICD-C. Diese finden Sie (nur über Uni-Rechner erreichbar) unter
  `https://wcc-web.informatik.uni-ulm.de/lehre/CfES/ICD-C`
  Öffnen Sie im links erscheinenden Menü den Punkt *Data Structures*, und Sie erhalten detaillierte Informationen zu sämtlichen Klassen und Methoden von ICD-C.