

## Übungsblatt 09

Termin: 2007/06/05

**Ü 9.1 Zwischencode für Boolesche Ausdrücke, Backpatching**

Gegeben sei der Boolesche Ausdruck:

```
not ( a < b or a < c )
```

Der von diesem Ausdruck erzeugte Drei-Adress-Code soll zur Sprungmarke `Ltrue` springen, falls der Wert des Ausdrucks `true` ergibt, sonst zur Sprungmarke `Lfalse`.

- Übersetzen Sie den Ausdruck in Drei-Adress-Code gemäß der syntaxgesteuerten Definition auf VO-Folie 8.21 (Kontrollflussübersetzung), und geben Sie den annotierten Ableitungsbaum an.
- Wie a) für das Übersetzungsschema auf VO-Folie 8.25 (Backpatching), wobei die Drei-Adress-Befehle in einem Quadrupelarray beginnend bei Index 100 abgelegt werden sollen.
- Warum wird Backpatching verwendet?

**Ü 9.2 Registerzuordnung durch Graphenfärbung**

Gegeben sei folgende Drei-Adress-Befehlssequenz:

```
a := b + c
e := a - d
d := d - b
b := d + c
```

- Eine Variable  $v$  heißt lebendig an der Stelle  $i$  einer Befehlssequenz, wenn  $v$  in einem möglichen Ausführungspfad *nach* der Ausführung des Befehls  $i$  verwendet (d.h. ausgelesen) wird und inzwischen nicht verändert worden ist. Bestimmen Sie alle lebendigen Variablen an jeder Stelle der gegebenen Befehlssequenz unter der Annahme, dass an der Stelle des letzten Befehls der Sequenz nur die Variablen `b` und `e` lebendig sind.
- Bestimmen Sie den Registerinterferenzgraphen der gegebenen Befehlssequenz, der mögliche Konflikte bei der Registerzuordnung der Variablen darstellt. Die Variablen bilden die Knoten, und eine Kante verbindet zwei Knoten, wenn die eine Variable lebendig an jener Stelle ist, an der die andere Variable als L-Wert einer Zuweisung auftritt. Benachbarte Knoten dürfen also nicht demselben Register zugeordnet werden.
- Angenommen, der gegebene Drei-Adress-Code soll in eine Zielsprache für eine Maschine mit 3 allgemein verwendbaren Registern übersetzt werden, wobei möglichst alle Variablen in Registern gehalten werden sollen. Das Registerzuordnungsproblem ist dann äquivalent zum Problem der Färbung des Registerinterferenzgraphen aus b) mit 3 Farben: alle Knoten sollten so gefärbt werden, dass benachbarte Knoten jeweils verschiedene Farben erhalten. Eine heuristische, rekursive Methode für die Graphenfärbung mit  $k$  Farben ist:
  - Entferne aus dem Graphen  $G$  einen Knoten  $v$ , der weniger als  $k$  Nachbarn hat, und alle seine Kanten. Der resultierende Graph sei  $G'$ .
  - Führe die Graphenfärbung für  $G'$  rekursiv durch, falls  $G'$  mehr als  $k$  Knoten hat. Sonst ist die Graphenfärbung für  $G'$  trivial.

(3) Füge den Knoten  $v$  und seine Kanten wieder zu  $G'$  hinzu und wähle für  $v$  eine Farbe, die nicht schon seinen Nachbarn zugeordnet ist.

Finden Sie eine Registerzuordnung für die gegebene Befehlssequenz mit Hilfe der Graphenfärbungsmethode.

d) Welche Aktion des Codegenerators ist nötig, falls in einem Rekursionsschritt der Graphenfärbung alle Knoten mindestens  $k$  Nachbarn haben?