

Stand der Studienarbeit *Typinferenz in Java I*

Die Theoriephase des 5. Semesters bestand aus Sicht der Studienarbeit hauptsächlich aus Einlesen in die gegebene Literatur (Studienarbeit von Jörg Bäuerle und Ausarbeitung von Martin Plümicke) und dem Erstellen von Use-Cases für die bestehende Implementierung des Typinferenz-Algorithmus. Viele Use-Cases wurden korrekt abgehandelt, ein Paar jedoch führten auf Fehler, die bei der Vervollständigung der aktuellen Version zur β -Version behoben werden müssen. Bei den anderen Aufgaben, die zu erledigen waren muss zuerst die Grammatik modifiziert werden. Der Algorithmus muss anschliessend geändert werden.

Bisher gefundene Bugs

return in while oder if

Bei einer Programmsequenz folgender Art erzeugt der Typinferenzalgorithmus einen Laufzeitfehler (**Block.TRStatements(): Unerreichbare Anweisung**):

```
if(i>1) {
    return(">1");
}
return("<=1");
```

Ansätze zur Lösung dieses Problems sind bereits vorhanden, die nötige SuperTyp-Unifikations-Methode ist jedoch noch nicht implementiert.

Überladene Methoden

Überladene Funktionen können bisher vom Typinferenzalgorithmus bisher auch nicht erkannt werden. Dies liegt an der Vorgehensweise beim Abspeichern der Methoden in die Hashstruktur. Dort wird lediglich gespeichert, **wie** die Methode heisst und **wieviele** Parameter sie hat. Taucht eine Methode mit den selben Eigenschaften auf, wird die erste überschrieben und demnach nicht bearbeitet.

```
public String overloaded(a){
return(a);
}
public int overloaded(a){
return(a);
}
```

Lösen könnte man das Problem durch das Anpassen des Hash-Schlüssels oder das Einführen einer neuen Datenstruktur (p.ex. Vector). Jedoch muss die prinzipielle Lösbarkeit dieses Problems in der bisherigen Implementierung zuerst geprüft werden.

Methode in *if-Statement*

Bei folgendem Konstrukt tritt momentan ein Fehler auf (**IfStmt.TRStatement(): Bedingung muss boolean sein!**). Die Ursache ist jedoch noch nicht vollständig geklärt.

```
boolean simple2(){
return(1>6);
}
simple3(){
    if(simple2()){
    }
}
```