

Ziel in dieser Woche ist die Entwicklung der semantische Prüfung. Die semantische Analyse holt Dinge nach, die wegen der beschränkten Grammatikklassen nicht während des Parsens erledigt werden können. Dazu gehören Namensanalyse und Typprüfung.

Im Allgemeinen gilt: Liefert diese Prüfung keine Fehler so muss der Compiler (bis auf Ressourcenbeschränkungen) in der Lage sein das Programm zu übersetzen.

Aufgabe 1: Namensanalyse

- Welche Elemente im AST müssen verlinkt werden? Welche Namensräume gibt es? Wie sind die Namensräume verschachtelt?
- Entwerfen Sie eine Schnittstelle zur Namensanalyse während eines AST-Durchlaufs und implementieren Sie diese.
- Benutzen Sie ihre Namensanalyse um Deklarationen mit ihren Benutzern zu verbinden und legen Sie die Ergebnisse im AST ab.

(Wir empfehlen sich die Vorlesungsfolien und Übungsblatt 8 der Vorlesung „Sprachtechnologie und Compiler“ aus dem Sommersemester 2012 anzusehen).

Aufgabe 2: Semantische Prüfung – Typen

- Welche Typen existieren in MiniJava? Braucht man weitere Typen?
- Stellen Sie Typisierungsregeln für die Operationen in MiniJava auf.
- Implementieren Sie diese Typisierungsregeln.

Aufgabe 3: Semantische Prüfung – Sonstiges

- Welche weiteren Tests sind in der Semantikphase nötig?
- Implementieren Sie diese!

Aufgabe 4: Testen

Implementieren Sie einen Schalter `--check` in ihrem Compiler, der die angegebene Datei parst und anschließend die semantische Analyse durchführt. Fehlerhafte Programme sollten abgelehnt werden. Der Compiler sollte dazu eine Meldung ausgeben in der die Zeichenkette `error` vorkommt und das Programme sollte mit einem Rückgabewert ungleich 0 beendet werden (`System.exit(1);`).

Die fertigen Compiler die `--print-ast` (von Blatt 4) und `--check` implementieren, sollten Sie bei uns bis zum 20.11. abgeben.