

1. Übungsblatt

Ausgabe: 14.04.16

Abgabe: 21.04.16

In diesem Übungsblatt machen wir unsere ersten praktischen Erfahrungen mit der Sprache C0 und der Implementierung der zugehörigen Werkzeuge.

Dazu laden Sie sich das Rahmenwerk für das C0-Analyse-Werkzeug (*C0 Analysing Tool* oder *cat*¹) von der Webseite herunter.

1.1 Einschränkungen und Umformungen

5 Punkte

Erweitern Sie *cat* um eine Phase, welche die folgenden Einschränkungen für erlaubte C0-Programme prüft:

- Zuweisungen erscheinen nur auf oberster Ebene (alle Ausdrücke sind seiteneffektfrei);
- Es werden keine strukturierten Typen (**struct**) oder dazugehörige Operationen (Feldselektion) verwendet;
- Es werden keine Zugriffe auf Arrays, Dereferenzierung, oder der Addressoperator verwendet;
- Die Ausdrücke im ersten und dritten Argument der **for**-Schleifen sind Zuweisungen oder Funktionsaufrufen;
- Ferner sollte diese Phase **for**-Schleifen in **while**-Schleifen umformen, nach folgendem Muster:

```
for (init; cond; incr) body ==> init; while(cond) { body; incr }
```

1.2 Statische Analyse

15 Punkte

Erweitern Sie *cat* um eine statische Analysephase (Typechecker).

Nach dem Parsen haben alle Bezeichner (Identifier) den leeren Typ (DummyType). Die statische Analyse sollte rekursiv den Syntaxbaum traversieren, und für jeden Bezeichner den deklarierten Typ eintragen (bzw. einen Fehler ausgeben, wenn der Bezeichner nicht deklariert ist).

Ferner sollte die statische Analyse folgende Bedingungen prüfen:

- Haben alle binären und unären Operatoren die richtigen Typen?
- Haben alle Funktions- und Prozeduraufrufe den richtigen Typen?
- Stimmt der Typ eines **return**-Statements mit dem Ergebnistyp der Funktion überein?
- Haben alle Spezifikationen den richtigen Typ (nämlich IntType)?
- Werden keine lokalen (einschließlich Funktionsparameter) oder globale Variablen doppelt deklariert?

Hinweise:

1. Die Klasse *Env* stellt einen Datentyp bereit, in dem lokale und globale Deklarationen verwaltet werden.
2. Strukturieren Sie Ihre Analyse wie folgt:
 - Eine Funktion

¹Unsere Marketingabteilung behauptete, „Katzen gehen im Internet immer“, deshalb der Name.

def checkExpr(ctxt: Env, e: Expr): Option[(Expr, Type)] = ???

welche *Ausdrücke* analysiert, und einen voll typannotierten Ausdruck und den Typen dieses Ausdrucks zurückgibt;

- darauf aufbauend eine Funktion

def checkStmt(ctxt: Env, s: Stmt): Option[Stmt] = ???

welche eine *Anweisung* analysiert, und annotiert zurückgibt;

- darauf aufbauend eine Funktion

def checkExtDecl(ctxt: Env, d: ExtDecl): Option[(Identifier, Type)] = ???

welche eine *externe Deklaration* (globale Deklaration von Funktionen oder Variablen, oder Funktionsdefinitionen) analysiert, und einen Tupel aus dem deklarierten Bezeichner und dessen Typ zurückgibt;

- und zum Schluß eine Funktion

def analyse(tu: TranslationUnit): Option[Env] = ???

welche eine gesamte Datei analysiert, und als Environment zurückgibt.