

Aufgabe 1: AGs: Deklaration nach Benutzung

Gegeben folgende Grammatik:

$$\begin{aligned} S &\rightarrow \text{CompoundStatement} \\ \text{CompoundStatement} &\rightarrow \{ \text{StatementList} \} \\ \text{StatementList} &\rightarrow \text{Statement StatementList} \mid \varepsilon \\ \text{Declaration} &\rightarrow \text{Type id ;} \\ \text{Type} &\rightarrow \text{id} \\ \text{EStatement} &\rightarrow \text{Expression ;} \\ \text{Expression} &\rightarrow \text{id} \mid (\text{Expression}) \\ \text{Typedef} &\rightarrow \text{newtype id ;} \\ \text{Statement} &\rightarrow \text{Declaration} \mid \text{EStatement} \mid \text{Typedef} \mid \text{CompoundStatement} \end{aligned}$$

Erzeugen Sie ein System von Attributierungsregeln die erlauben Typen und Variablen zu benutzen, die erst später deklariert werden.

Aufgabe 2: Praxis: Namensanalyse

Unter <http://pp.info.uni-karlsruhe.de/lehre/WS201011/compiler/uebung/intern/nameana.zip> befindet sich Java Sourcecode mit einem Parser und Interpreter. Die implementierte Sprache besitzt eine Zuweisungs- und einer Ausgabeoperation. Ausserdem können mit { und } Namensbereiche geschachtelt werden. Die Verwaltung der Namenstabelle in NameTable.java wurde entfernt.

- Implementieren Sie die fehlende Funktionalität in NameTable.java

Beispiel für eine Eingabe:

```
{
    foo = "bar";
    print(foo);

    {
        foo = "bar2";
        print(foo);
    }

    print(foo);
}
```

Korrekte Ausgabe:

```
foo is bar
foo is bar2
foo is bar
```

Aufgabe 3: Java Bytecode

Eine detailliert Beschreibung der Java Virtual Machine (und Java Bytecode) finden Sie in der „The Java™ Virtual Machine Specification“.

3.1 Übersetzen

Übersetzen Sie den Ausdruck `this.x = 32 - this.foo() [2]` in Java Bytecode, geben Sie auch einen passenden Konstantenpool an. (`foo` ist eine nicht-statische Methode.

3.2 Wissen

- Was ist der Unterschied zwischen `aload 2` und `aload_2`?
- Mit den Befehlen `bipush` und `sipush` kann man 8 oder 16 Bit Immediate Werte auf den Stack legen. Warum gibt es keine Variante für 32 Bit Zahlen, was tut man stattdessen?
- Was ist der Unterschied zwischen `ret` und `return`?

3.3 Fehlersuche

Das folgende Java Bytecode Programm soll ABCDEFGHIJKLMNOPQRSTUVWXYZ ausgeben. Leider haben sich einige Fehler eingeschlichen, beschreiben und beheben Sie diese!

```
.class public print
.super java/lang/Object

.method public static main([Ljava/lang/String;)V
    .limit stack 5
    bipush 65          ; start with 65 ('A')

loop:
    ; use System.out.append() to print the character
    ; Signature: "PrintStream PrintStream.append(char c)"
    getstatic java/lang/System/out Ljava/io/PrintStream;
    invokevirtual java/io/PrintStream/append(C)Ljava/io/PrintStream;

    iconst_1
    iadd              ; increment

    bipush 90         ; if greater than 90 ('Z') jump to end
    if_icmpgt end

    goto loop        ; next loop iteration

end:
    ; call System.out.println() to produce a linebreak
    ; Signature: "void PrintStream.println()"
    getstatic java/lang/System/out Ljava/io/PrintStream;
    invokevirtual java/io/PrintStream/println()V
    return
.end method
```

Aufgabe 4: Speicherlayout

Bestimmen Sie ein Speicherlayout für die folgenden Strukturen, Unions und Klassen. Geben Sie dazu für jedes Feld einen Offset und für jede Struktur Größe und Alignment an! Die Beispiele sind C++ Programme.

```
struct Foo {
    int a;
    double b;
};
struct Bar {
    char c[3];
    short z;
};
union blup {
    Foo foo;
    Bar bar;
};
class Cls {
private:
    char c;
public:
    Bar bar;

    void print(void) { printf("%c\n", c); }

    char d;
    blup b;
    char z[1];
};
struct baz {
    Cls classes[11];
    int z;
    char c;
};
```