



Universität Karlsruhe (TH)

Lehrstuhl für Programmierparadigmen

Sprachtechnologie und Compiler WS 2008/2009

Dozent: Prof. Dr.-Ing. G. Snelting

Übungsleiter: Matthias Braun

<http://pp.info.uni-karlsruhe.de/>

snelting@ipd.info.uni-karlsruhe.de

braun@ipd.info.uni-karlsruhe.de

Übungsblatt 8

Ausgabe: 14.12.2008

Besprechung: 17.12.2008

Aufgabe 1: AGs: Deklaration nach Benutzung

Gegeben folgende Grammatik:

```
S → CompoundStatement
CompoundStatement → { StatementList }
StatementList → Statement StatementList
Declaration → Type id ;
Type → id ;
EStatement → Expression ;
Expression → id | ( Expression )
Typedef → newtype id ;
Statement → Declaration | EStatement | Typedef | CompoundStatement
```

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

Aufgabe 2: Aufrufkonventionen

Beschreiben Sie die Aufrufkonventionen (Calling Conventions) von 2 der folgenden Prozessorarchitekturen auf einem Betriebssystem Ihrer Wahl:

1. IA-32
2. Alpha
3. PowerPC
4. IA-64
5. ARM

Suchen Sie hierzu im Internet nach der entsprechenden Dokumenten im Internet (z.B. auf den Herstellerseiten).

Geben Sie für jede der von Ihnen gewählten Plattformen an, wie die Argumente beim Aufruf folgender C-Funktion übergeben werden. Wie wird der Rückgabewert zurück gegeben?

```
struct test_struct {
    int x;
    int y;
};

int foobar(int i, char c, double d, int j, struct test_struct s) {
    return 42;
}
```

Aufgabe 3: Activation Records

Gegeben folgendes Programm:

```
int iterations ;

int fib(int x)
{
    int r;

    iterations++;
    if (x == 0) {
        r = 0;
    } else if (x == 1) {
        r = 1;
    } else {
        r = fib(x-1) + fib(x-2);
    }
    return r;
}

int main(void)
{
    int input;
    scanf("%d", &input);
    printf("Result: %d, needed %d iterations\n", fib(input), iterations );
    return 0;
}
```

Stellen Sie sich einen nicht optimierenden Übersetzer vor, der alle Variablen auf dem Stack ablegt. Ausserdem verlangt die Aufrufkonvention dass alle Argumente auf dem Stack übergeben werden. Der Prozessor selbst legt beim betreten einer Funktionen die Rücksprungadresse auf den Stack.

Stellen Sie die Struktur des Stacks unmittelbar vor dem Aufruf der Funktion fib dar und nach dem 2. Rekursionsschritt der Funktion fib.