

Programmiersprachen und Übersetzer

Übung 8

Ausgabe am: **4. Juni 2024**
Abgabe bis: **11. Juni 2024**

Aufgabe 1 - Type Inference (4 Punkte)

(a) Bestimmen Sie, unter Benutzung der bekannten Typregeln für λ_{\rightarrow} sowie des $+$ -Operators,

$$\frac{(x : T) \in \Gamma}{\Gamma \vdash x : T} \text{ (T-VAR)}$$

$$\frac{\Gamma, x : T \vdash t : T'}{\Gamma \vdash (\lambda x : T. t) : T \rightarrow T'} \text{ (T-ABS)}$$

$$\frac{\Gamma \vdash t_1 : T \rightarrow T' \quad \Gamma \vdash t_2 : T}{\Gamma \vdash t_1 \ t_2 : T'} \text{ (T-APP)}$$

$$\frac{\Gamma \vdash t_1 : Int \quad \Gamma \vdash t_2 : Int}{\Gamma \vdash t_1 + t_2 : Int} \text{ (T-PLUS)}$$

den Typen des folgenden SML-Ausdrucks

```
fn x => fn y => y(x) + 1;
```

durch Aufstellen der *type constraints* und Lösen des Gleichungssystems per Unifikation.

Die Abgabe soll in einer allgemein lesbaren Datei **aufgabe1.*** erfolgen.

Aufgabe 2 - Kontrollflussgraphen (5 Punkte)

Gegeben ist der x86-Assembler Code in **aufgabe2.s**. Bilden Sie den funktionslokalen Kontrollflussgraphen (CFG) mit maximalen Basisblöcken.

$$\text{fn } u \Rightarrow \text{fn } y \Rightarrow y(u) + 1$$

the above Function is $\equiv \lambda u. (\lambda y. (y u + 1))$
 $(\lambda u. (\lambda y. (y u : \alpha_1 + 1) : \alpha_2) : \alpha_3) : \alpha_4$

Constraint Generation:

GV
↑

$$\begin{aligned} \llbracket \alpha_4 \rrbracket &= \llbracket u \rrbracket \rightarrow \llbracket \alpha_3 \rrbracket \\ \llbracket \alpha_3 \rrbracket &= \llbracket y \rrbracket \rightarrow \llbracket \alpha_2 \rrbracket \\ \llbracket \alpha_2 \rrbracket &: \text{int} \\ \llbracket \alpha_1 \rrbracket &: \text{int} \\ \llbracket y \rrbracket &= \llbracket u \rrbracket \rightarrow \llbracket \alpha_1 \rrbracket \end{aligned}$$

Rules

Stack

Substitutions

$$\begin{aligned} \llbracket \alpha_4 \rrbracket &= \llbracket u \rrbracket \rightarrow \llbracket \alpha_3 \rrbracket \\ \llbracket \alpha_3 \rrbracket &= \llbracket y \rrbracket \rightarrow \llbracket \alpha_2 \rrbracket \\ \llbracket \alpha_2 \rrbracket &: \text{int} \\ \llbracket \alpha_1 \rrbracket &: \text{int} \\ \llbracket y \rrbracket &= \llbracket u \rrbracket \rightarrow \llbracket \alpha_1 \rrbracket \end{aligned}$$

3

$$\begin{aligned} \llbracket \alpha_3 \rrbracket &= \llbracket y \rrbracket \rightarrow \llbracket \alpha_2 \rrbracket \\ \llbracket \alpha_2 \rrbracket &: \text{int} \\ \llbracket \alpha_1 \rrbracket &: \text{int} \\ \llbracket y \rrbracket &= \llbracket u \rrbracket \rightarrow \llbracket \alpha_1 \rrbracket \end{aligned}$$

$$\llbracket \alpha_4 \rrbracket \mapsto (\llbracket u \rrbracket \rightarrow \llbracket \alpha_3 \rrbracket)$$

3

$$\begin{aligned} \llbracket \alpha_2 \rrbracket &: \text{int} \\ \llbracket \alpha_1 \rrbracket &: \text{int} \\ \llbracket y \rrbracket &= \llbracket u \rrbracket \rightarrow \llbracket \alpha_1 \rrbracket \end{aligned}$$

$$\llbracket \alpha_4 \rrbracket \mapsto (\llbracket u \rrbracket \rightarrow (\llbracket y \rrbracket \rightarrow \llbracket \alpha_2 \rrbracket))$$

$$\llbracket \alpha_3 \rrbracket \mapsto (\llbracket y \rrbracket \rightarrow \llbracket \alpha_2 \rrbracket)$$

3

$$\begin{aligned} \llbracket \alpha_1 \rrbracket &: \text{int} \\ \llbracket y \rrbracket &= \llbracket u \rrbracket \rightarrow \llbracket \alpha_1 \rrbracket \end{aligned}$$

$$\llbracket \alpha_4 \rrbracket \mapsto (\llbracket u \rrbracket \rightarrow (\llbracket y \rrbracket \rightarrow \llbracket \alpha_2 \rrbracket))$$

$$\llbracket \alpha_3 \rrbracket \mapsto (\llbracket y \rrbracket \rightarrow \llbracket \alpha_2 \rrbracket)$$

$$\llbracket \alpha_2 \rrbracket : \text{int}$$

3

$$\llbracket y \rrbracket = \llbracket u \rrbracket \rightarrow \text{int}$$

$$\llbracket \alpha_4 \rrbracket \mapsto (\llbracket u \rrbracket \rightarrow (\llbracket y \rrbracket \rightarrow \text{int}))$$

$$\llbracket \alpha_3 \rrbracket \mapsto (\llbracket y \rrbracket \mapsto \text{int})$$

$$\llbracket \alpha_2 \rrbracket : \text{int}, \llbracket \alpha_1 \rrbracket : \text{int}$$

3

$$\llbracket \alpha_4 \rrbracket \mapsto (\llbracket x \rrbracket \rightarrow ((\llbracket x \rrbracket \rightarrow \text{int}) \rightarrow \text{int}))$$

$$\llbracket \alpha_3 \rrbracket \mapsto ((\llbracket x \rrbracket \rightarrow \text{int}) \rightarrow \text{int})$$

$$\llbracket \alpha_2 \rrbracket : \text{int}$$

$$\llbracket \alpha_1 \rrbracket : \text{int}$$

$$\llbracket y \rrbracket \mapsto (\llbracket x \rrbracket \rightarrow \text{int})$$

$$\llbracket x \rrbracket \rightarrow ((\llbracket x \rrbracket \rightarrow \text{int}) \rightarrow \text{int})$$

Die Abgabe soll in einer allgemein lesbaren Datei `aufgabe2.*` erfolgen.

Aufgabe 3 - Kurzschlussauswertung (3 Punkte)

Kurzschlussauswertung zeichnen sich dadurch aus, dass ihre Ausführung möglichst früh abbricht. Ein Beispiel in C ist der Ausdruck `a && b && c`.

Schreiben Sie ein Programm in der Intermediate Representation aus der Vorlesung, das das Verhalten des angefügten C-Codes imitiert. Sie müssen lediglich den Funktionskörper implementieren und können die Variablen `a`, `b`, `c` direkt benutzen.

```
int func(int a, int b, int c) {  
    return a && b && c ? 1 : 0;  
}
```

Die Abgabe soll in der Datei `aufgabe3.ir` geschehen.

Abgabe

Die Antworten müssen im bestehenden Repository **in einem neu zu erstellenden Verzeichnis "ex8"** eingereicht werden.