

# Programmiersprachen und Übersetzer Übung 9

Ausgabe am: **Juni 11 2024**

Abgabe bis: **Juni 18 2024**

**Abgabe:** Die Antworten müssen im bestehenden Repository **in einem neu zu erstellenden Verzeichnis "ex9"** eingereicht werden.

Wir betrachten ein *simply-typed  $\lambda$  calculus* mit Subtypisierung, mit den Typen  $A$ ,  $B$ ,  $C$ ,  $D$ , und  $E$ , sowie den folgenden Subtypisierungsregeln:

$A <: B$     *A ein Subtyp von B ist*  
 $B <: C$     *B " " " C "*  
 $D <: E$     *D " " " E "*  
 $A <: E$     *A " " " E "*

## Aufgabe 1 - Subtyping (5 Punkte)

Geben Sie alle Paare  $S_i$  und  $T_j$  an, für die  $S_i <: T_j$  gilt.

*$D \rightarrow A$      $D <: A$*

$S_1 = \{x : D\} \rightarrow \{y : A\}$

$T_1 = E$

$S_2 = \{y : B\}$

$T_2 = \{x : D\}$

$S_3 = \{x : A, y : A\}$

$T_3 = \{\}$

$S_4 = D$

$T_4 = \{x : E\} \rightarrow \{y : B\}$

$S_5 = \{x : B\}$

$T_5 = \{y : E\}$

# Aufgabe 1.

S<sub>1</sub>: Es gibt kein  $\overline{1j}$  ;  $S_1 <: \overline{1j}$

S<sub>2</sub>:  $S_2 <: \overline{13}$

$$\frac{}{S_2 = \{y:B\} <: \{y=\overline{13}\}} \text{ (S-Redwidth)}$$

S<sub>3</sub>:  $S_3 <: \overline{13}$

$S_3 <: \overline{15}$

$$\frac{A <: E}{\text{ (S-RedDepth)}}$$

$$\frac{\{y:A\} <: \{y:E\}}{\text{ (S-Redwidth)}}$$

$$\frac{\{y:A, u:A\} <: \{y:E\}}{\text{ (S-RedForm)}}$$

$$\frac{}{S_3 = \{u:A, y:A\} <: \{y:E\} = \overline{15}}$$

S<sub>4</sub>:  $S_4 <: \overline{11}$

$$S_4 = D <: E = \overline{11}$$

S<sub>5</sub>:  $S_5 <: \overline{13}$

$$\frac{}{S_5 = \{u:B\} <: \{y=\overline{13}\}} \text{ (S-Redwidth)}$$

## Aufgabe 2 - Joins und Meets (6 Punkte)

Unter Berücksichtigung der Subtypisierungsregeln:

- A type  $J$  is called a join of a pair of types  $S$  and  $T$  if  $S <: J$ ,  $T <: J$ , and, for all types  $U$ , if  $S <: U$  and  $T <: U$ , then  $J <: U$ .
- A type  $M$  is a meet of  $S$  and  $T$  if  $M <: S$ ,  $M <: T$ , and, for all types  $L$ , if  $L <: S$  and  $L <: T$ , then  $L <: M$ .

Geben Sie die "joins" und "meets" der Paare von Typen an, sofern sie existieren, andernfalls *nicht definiert*. Wir nehmen an, dass wir einen *Top* Typen haben, zu dem jeder anderer Typ ein Subtyp ist.

$A <: B$        $D <: E$   
 $B <: \_$        $A <: E$

a)  $\{x : E, y : C\}$  and  $\{x : B, y : B\}$

join =  $\{x : \text{Top}, y : C\}$

meet =  $\{x : E, y : B\}$

b)  $\{x : B\}$  and  $\{y : B\}$

join =  $\{\}$

meet =  $\{x : B, y : B\}$

c)  $\{x : A\}$  and  $\{x : D\}$

join =  $\{x : E\}$

meet = Es gibt keine Subtypen zu  $A$  und  $D$ . Deshalb ist meet nicht definiert.

## Aufgabe 3 - Vtables (2 Punkte)

Gegeben sind die folgenden Klassendeklarationen:

```
class One {
    void setTag(Tag t) { ... }
    Tag getTag() { ... }
}

class Two extends One {
    @Override
    void setTag(Tag t) { ... }
    void reset() { ... }
}
```

Geben Sie die (vollständig materialisierten) vtables für **One** und **Two** an.

# variable für One

method	address
setTag(Tag t) v	&One.setTag(Tag t) v
getTag()	&One.getTag()

method	Address
setTag(Tag t) v	&Two.setTag(Tag t) v
getTag() Tag	&One.getTag() Tag
reset() v	&Two.reset() v