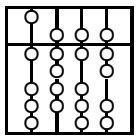
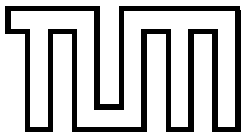


TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK



Lehrstuhl II *Sommersemester 2007*
Prof. Helmut Seidl *Andrea Flexeder*

Compilerbau

Aufgabe 35a)

Die Funktionsdefinition:

```
| ?- Af = to(Ax,Fx), Af = to(Fx,FFx), Fn = to(tuple(Af,Ax),FFx).
```

Allgemeinster Unifikator:

```
    Af = to(Ax,Ax)
    FFx = Ax
    Fn = to(tuple(to(Ax,Ax),Ax),Ax)
    Fx = Ax
```

Ein konkreten Aufruf:

```
| ?- Tinc = to(int,int), Ttwice = to(tuple(to(A,A),A),A),
     Ttwice = to(Tincfour,Ttwiceincfour),
     Tincfour = tuple(Tinc,int), Av = Ttwiceincfour.
```

Allgemeinster Unifikator:

```
    A = int
    Av = int
    Tinc = to(int,int)
    Tincfour = tuple(to(int,int),int)
    Ttwice = to(tuple(to(int,int),int),int)
    Ttwiceincfour = int
```

Aufgabe 35b)

Die Funktionsdefinition:

```
| ?- Amap = Tfn, Tfn = to(tuple(Af,Al),Tcase),
     Tcase = list(XA), Tcase = Te2, Al = list(Ax),
     Al = Axs, Te2 = Amapfxs, Te2 = list(Tfx),
     Af = to(Ax,Tfx), Amap = to(tuple(Af,Axs),Amapfxs).
```

Allgemeinster Unifikator:

```
    Af = to(Ax,Tfx)
    Al = list(Ax)
    Axs = list(Ax)
    Tcase = list(Tfx)
    Te2 = list(Tfx)
    Tfn = to(tuple(to(Ax,Tfx),list(Ax)),list(Tfx))
    Amap = to(tuple(to(Ax,Tfx),list(Ax)),list(Tfx))
    Amapfxs = list(Tfx)
    XA = Tfx
```

Aufgabe 36)

Die Funktionsdefinition:

```
| ?- Ae = Tfnl, Tfnl = to(Al,Tfny), Tfny = to(Ay,Tcase),
     Tcase = bool, Tcase = Tif, Al = list(Ax), Al = Axs,
     Teq = to(Beta,to(Beta,bool)), Teq = to(Ax,to(Ay,bool)),
     Tif = bool, Tif = Tecall, Ae = to(Axs,to(Ay,Tecall)).
```

Allgemeinster Unifikator:

```
Ae = to(list(Ax),to(Ax,bool))
Al = list(Ax)
Axs = list(Ax)
Ay = Ax
Beta = Ax
Tcase = bool
Tecall = bool
Teq = to(Ax,to(Ax,bool))
Tfnl = to(list(Ax),to(Ax,bool))
Tfny = to(Ax,bool)
Tif = bool
```