Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Einführung in das funktionale Programmieren:

Sommersemester 1997

Übungsblatt Nr. 6

Aufgabe 1

Auf Blatt 4 und Blatt 5 wurden die Funktionen `nextRedex` und `reduce` als Aufgaben programmiert. Verwenden Sie diese Funktionen, um Funktionen zu schreiben, die einen `KFPexp`-Ausdruck zu WHNF bzw. zu NF auswerten.

Aufgabe 2

In der Vorlesung wurde **KFP** erweitert. Unter anderem kam `letrec` hinzu. Wir erweitern also `KFPexp` entsprechend:

```
data KFPdefn = (String, KFPexp)
type KFPtype = (String, [(String, Int)])
data KFPexp = VarConst String
            | App KFPexp KFPexp
            | Case KFPtype KFPexp [KFPexp]
            | Letrec [KFPdefn] KFPexp
```

Um `letrec` wieder in das ursprüngliche **KFP** zurückzuübersetzen wird λ-Lifting verwendet. Dabei wird z.B. ein **KFP**-Programm:

```
h p q = p*q
f x = letrec g y = (h x y):(g y) in g 3
main = f 6
```

zu

```
h p q = p*q
$g g x y = (h x y):(g y)
f x = letrec g = $g g x in g 3
main = f 6
```

Beim λ-Lifting ist es nötig zu jedem Ausdruck, die in ihm vorkommenden freien Variablen zu bestimmen, um diesen dann eindeutige Namen zu geben, deswegen statten wir `KFPexp`-Ausdrücke mit Annotationen aus, in denen wir die im Ausdruck auftretenden freien Variablen sammeln.

```
data KFPdefnAnn = (String, KFPexpAnn)
type KFPtypeAnn = (String, [(String, Int)])
data KFPexpAnn = ([String], KFPexpAnn')
data KFPexpAnn' = VarConst String
            | App KFPexpAnn KFPexpAnn
            | Case KFPtype KFPexpAnn [KFPexpAnn]
            | Letrec [KFPdefnAnn] KFPexpAnn
```

Schreiben Sie eine Funktion `freeVarsExp`, die aus einem `KFPexp`-Ausdruck einen `KFPexpAnn`-Ausdruck macht, in dem die Teilausdrücke entsprechend annotiert sind.

Wenn `KFPprog` deklariert ist als `type KFPprog = [(String, [String], KFPexp)]` bzw. `type KFPprogAnn = [(String, [String], KFPexpAnn)]`, wie sieht dann `freeVars` aus, die aus einem `KFPprog` in ein entsprechendes annotiertes `KFPprogAnn` macht.

(wird fortgesetzt)