

Dan V. Popa



```
qsort [] = []
qsort (x:xs)
  = qsort pre ++ [x] ++ qsort post
  where
    pre  = [ n | n<- xs, n < x ]
    post = [ n | n<- xs, n >= x ]
```

Cuprins

Cuvânt înainte ; Foreword.....	11
Introducere	15
1. Funcții și aritmetică	17
1.1. Tipurile Integer, Bool, Char și altele; Tipurile compuse, liste și perechi	20
1.2. Operatorii (operațiile).....	23
1.3. Funcțiile. Un prim exemplu , funcția increment	26
1.4. Rularea unui program cu hugs pe un sistem Linux.....	29
1.5. Despre variabilele folosite la semnăturile funcțiilor și semnul ->	31
1.6. Suma a două numere.....	32
2. Pe urmele funcțiilor care manipulează funcții.....	36
2.1. Funcții care manipulează funcții: Sumarea; Funcția care aplică o altă funcție de două ori.....	38
2.2. Manipularea repetată a funcțiilor: Recursivitatea;	39
Exemple: Factorialul și funcția lui Ackerman-Peter.....	40
2.3. Operatorul "sumă" - o primă încercare nereușită (deoarece Haskell 98 nu permite variabile ale căror nume se repetă în același șablon)	44
2.4. Pattern-uri și pattern-uri cu gardă Sign – semnul unui număr	47
2.5. Sigma refăcut	49
2.6. Factorialul refăcut.....	51
2.7. Despre o sumă de zerouri sau despre lambda expresii și despre șablonul pentru orice (_).....	54

2.8.Cum putem folosi lambda expresiile.....	59
2.9. Lambda expresiile cu parametri multipli	60
3. Cât mai multe despre tipuri și crearea lor.....	62
3.1. Un vechi exemplu: Triunghiul,declarația <i>data</i>	62
3.2.Tipuri utilizator și reuniunea	68
3.3.Tipuri recursive și arbori polimorfi	71
3.4.Liste în Haskell.....	76
3.5.Liste și iar liste, liste de caractere, stringuri	85
3.6. Tipuri sinonime, declarația <i>type</i>	91
3.7. Mulțimi implicite reprezentate ca liste.....	93
3.8. Secvențe aritmetice.....	96
4. Structuri de date infinite și alte noțiuni utile.....	98
4.1.Structuri de date infinite	98
4.2.Funcții care manipulează liste, extrase din Standard Prelude.....	108
4.3.CASE și IF în Haskell apoi LET...IN	113
4.4. Și ... WHERE ... , un fel de LET...IN... scris invers.....	121
5. Reprezentarea unui limbaj în Haskell.....	124
5.1. Sintaxa (abstractă)	125
5.1.1.Un mic interpretor	129
5.2.Cum exprimăm valorile rezultate	136
5.3.Cum reprezentăm memoria sau asocierea variabile-valori	138
5.4. Compunerea valorilor,evaluarea termenilor,pas și proces de calcul.....	140
5.5.Semantică și interpretare.....	142

6. Monada	145
6.1.Monada – scurtă prezentare (de interes istoric).....	148
6.2.O lucrare celebră și notațiile de atunci,	150
6.3.Funcția polimorfică "unitM".....	154
6.4.Funcția polimorfică "bindM".....	156
6.5.Transcrierea semanticii în do-notație.....	158
6.6.Operatori de lifting	160
6.7.Observații și concluzii	163
6.8 Un interpretor rescris și comentat, realizat cu o instanță a clasei Monad	164
7. 7 Monade utilizate de programatori.....	179
7.1. Monada identitate.....	179
7.2. Monada de I/O.....	180
7.3. Monada stărilor.....	182
7.4.Monada cu stări și (string de) I/O.....	186
7.5.Monada listelor.....	187
7.6. Monada Maybe.....	189
7.7.Monada parserelor.....	190
8.Studiu de caz: evaluatorul de expresii	194
9.Studiu de caz: generatorul de cod	222
9.1.Doar două vorbe despre lazy evaluation	222
9.2. Pregătim monada stărilor și formatul instrucțiunilor	224
9.3. Compilarea constantelor	232
9.4. Compilarea variabilelor	234

9.5. Compilarea declarațiilor variabilelor	236
9.6. Compilarea instrucțiunii <i>skip</i>	238
9.7. Compilarea instrucțiunii <i>read</i>	240
9.8. Compilarea instrucțiunii <i>write</i>	241
9.9. Compilarea instrucțiunii de atribuire	244
9.10. Compilarea operațiilor din expresii	245
9.11. Compilarea comparațiilor din expresii	249
9.12. Compilarea instrucțiunii condiționale (if)	250
9.13. Compilarea instrucțiunii structurate: secvența	254
9.14. Compilarea unei instrucțiuni ciclice: bucla while	256
9.15. Compilarea unei instrucțiuni ciclice: bucla do-while	259
9.16. Compilarea programului principal	262
9.17. Compilarea unui program principal: exemplul dat ca test	267
Anexa A : Instalarea Hugs98 pe o mașină Windows.....	270
Anexa B : Instalarea Hugs pe un sistem PCLinuxOS 9.....	280
Anexa C : Instalarea Hugs din distribuția Mandrake 10.0 pe o mașină Linux Mandriva 2005	283
Anexa D: Instalarea Happy pe o mașină Linux Mandriva 2005.....	284
Anexa E: Instalarea hugs pe sistemele Ubuntu	287
Anexa F: Proprietăți algebrice ale monadei parserelor - caz particular de monadă cu "plus" și "zerou".....	289
Anexa G : Schema interpretorului monadic	290
Anexa H: <i>Call by value</i> interpretorul (din finalul cap. 6) în do-notație	292
Anexa I: Întrebări de control	296
Bibliografie	302