

2.º Trabalho Prático

Métodos de Programação I
LESI/LMCC
Universidade do Minho
Ano Lectivo de 2001/2002

1 Introdução

É vulgar em documentos científicos (monografias, livros, artigos, *etc.*) existir um apêndice de bibliografia contendo a descrição de todos os documentos que foram citados ao longo do texto principal.

Um índice remissivo de autores é mais um apêndice ao texto principal indicando, por ordem alfabética de nomes de autores referidos e para cada nome de autor, a lista ordenada das páginas em que vem citadas publicações suas, por exemplo:

ARBIB — M. A.10, 11

BIRD — R.28

HOROWITZ — E.2, 3, 15, 16, 19

HUDAK — P.11, 12, 29

JONES — C. B.3, 7, 28

MANES — E. G.10, 11

SAHNI — S.2, 3, 15, 16, 19

SPIVEY — J.M.3, 7

WADLER — P.2, 3

No ambiente \LaTeX [3] para preparação de texto a geração de bibliografias pode ser feita automaticamente a partir de uma base de dados bibliográfica que associa a cada documento uma *chave de citação* que o identifica univocamente. O processamento de tal base de dados é realizado por um utilitário designado $\text{BIB}\text{\TeX}$.

Já a geração de índices remissivos de autores não está prevista na instalação \LaTeX de base. O objectivo deste trabalho é desenvolver um utilitário para esse efeito, a implementar e compilar em HASKELL [1, 2, 4].

Para perceber este trabalho será conveniente ter alguma convivência com \LaTeX . Em LINUX , faz parte da distribuição standard e é só experimentar. Em Windows, sugere-se a instalação de $\text{MiK}\text{\TeX}$ (www.miktex.org).

2 Material fornecido como “kit” para o projecto

Entre na página da disciplina e descarregue, descomprimindo-o (*eg.* via `unzip`), o ficheiro `mpi0102mp.zip` que contém o respectivo material pedagógico. Para além de outros ficheiros relevantes para a disciplina, deverá obter:

1. `mpi0102t2.lhs` - trata-se do ficheiro que está a ler neste momento, escrito em “*literate HASKELL*”. Isto significa que:

- se o carregar no HUGS, este interpretador lerá o código HASKELL nele contido e interpretá-lo-á. Sugestão: invoque o HUGS e experimente `:l mpi0102t2.lhs`.
 - se o processar via \LaTeX (ou \PDF\LaTeX) obterá este mesmo documento em *PDF*.
2. `mpi0102t2.sty` - trata-se de um ficheiro importado por `mpi0102t1.lhs` e contendo funções de processamento de texto expressas em sintaxe \LaTeX (como poderá ver inspeccionando este ficheiro, é normal programar *funcionalmente* em \LaTeX).
 3. `mpi0102t2.pdf` - trata-se do resultado do processamento de `mpi0102t2.lhs`, o ficheiro que está a ler.
 4. `mpi0102t2.bib` - trata-se de um ficheiro exemplo de bibliografia.
 5. `MonadicPComb.hs` - módulo auxiliar para reconhecimento do formato bibliográfico.

3 Análise do “kit” do projecto

Vamos usar o texto deste enunciado para ilustrar o funcionamento do processamento bibliográfico em \LaTeX :

1. Comece por invocar

```
pdflatex mpi0102t2.lhs
```

Verifique que à directoria de trabalho se acrescentaram vários ficheiros `mpi0102t2.*`, entre os quais aquele que está a ler neste momento — `mpi0102t2.pdf` — que poderá imprimir. Mas repare que há pontos de interrogação [?] nesse ficheiro, exactamente nos locais onde, em `mpi0102t2.lhs`, se invoca a função `ncite`.

2. Os argumentos da função `ncite` são chaves de referências bibliográficas que vai ser preciso cruzar com a base de dados de bibliografia — o ficheiro `mpi0102t2.bib` — que inclui registos como, por exemplo:

```
@book{Hu00,
  author = {P. Hudak},
  title = {\htmladdnormallink{
    The Haskell School of Expression - Learning Functional
    Programming Through Multimedia
  }}{
    http://www.cup.cam.ac.uk}
},
publisher = {Cambridge University Press},
edition = {1st},
year = {2000},
note = {ISBN 0-521-64408-9}
}
```

3. Para isso invoque `bibtex mpi0102t2`. Este utilitário deverá gerar o ficheiro `mpi0102t2.bbl` que irá ser anexado a `mpi0102t2.pdf`. De facto, se voltar a correr `pdflatex mpi0102t2.lhs` poderá reparar que uma nova secção de **Bibliografia** foi (automaticamente) anexada. Por exemplo, a entrada referente a Hu00 acima converte-se em

[2] P. Hudak. *The Haskell School of Expression - Learning Functional Programming Through Multimedia*. Cambridge University Press, 1st edition, 2000. ISBN 0-521-64408-9.

4. Repare ainda que `pdflatex mpi0102t2.lhs` gera um outro ficheiro auxiliar, `mpi0102t2.bkp`, cuja estrutura é muito simples, relacionando chaves de citação (argumentos de `ncite`) com a página onde essa citação é feita:

```
1,La86
1,J*99,Hu00,VB00
5,J*99,VB00
```

5. Os ficheiros `mpi0102t2.bkp` e `mpi0102t2.bib` contêm a informação necessária para gerar o índice de autores pretendido. É esse o objectivo do seu trabalho, a realizar em HASKELL.

Como é habitual nos trabalhos desta disciplina, fornece-se um “kit” para arranque do projecto. O enunciado concluir-se-á com sugestões para valorização do trabalho a realizar.

4 Utilização do “kit” do projecto

Após carregar `mpi0102t1.lhs` no seu interpretador de HASKELL, verifique que, após a importação de dois módulos que conhece,

```
import Mpi
import List
```

e de um outro mais à frente necessário à realização deste trabalho,

```
import MonadicPComb
```

se definem os tipos

```
type Bkp = [(Page,[Key])] -- bib.key/page data imported from *.bkp file
type Bib = [(Key,[Author])] -- bibtex data imported from *.bib file
```

correspondentes ao registo da informação desses ficheiros em HASKELL, onde

```
type Key = String
type Page = Int
data Author = Author initials :: String, surname :: String deriving Show
```

são tipos auxiliares.

Para efeitos de validação dispõe, no anexo C, duas constantes `bkp :: Bkp` e `bib :: Bib`, usando-se no segundo caso a abreviatura

```
x |-> y = (x,y)
```

Pode apreciar o uso dos combinadores ‘pointfree’ estudados nas aulas em funções como, por exemplo,

```
getb :: Bib -> [(String,[(String,String)])]
getb = map (id >< map (split initials surname))
```

que extrai informação relevante a partir de objectos do tipo `Bib`.

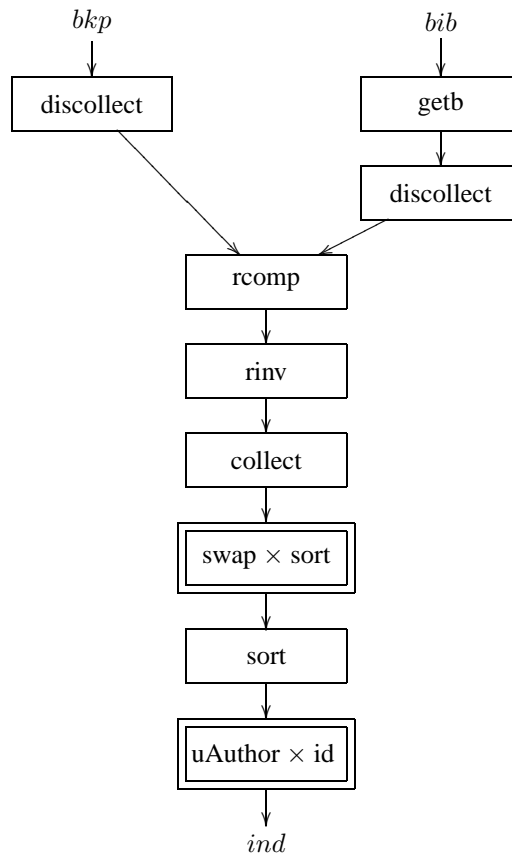
4.1 Estratégia

O nosso objectivo é sintetizar, a partir de `Bkp` e `Bib`, o índice de autores e páginas onde são referidos, tudo devidamente ordenado:

```
type Ind = [(Author,[Page])]
```

Será a partir do objecto do tipo `Ind` obtido que deverá ser produzido o ficheiro `mpi0102t2.ind` — por exemplo no formato sugerido na página 1 — que o \LaTeX anexará automaticamente ao PDF gerado através da ‘package’ `mmakeidx` (importada em `mpi0102t2.sty`).

Sugere-se a seguinte arquitectura para a função a realizar,



– em que

```
uAuthor = (uncurry Author) . swap
```

e \boxed{f} representa `map f` — que é fielmente captada pela seguinte expressão funcional ‘pointfree’:

```

mkInd :: Bkp -> Bib -> Ind
mkInd = curry (
  map (uAuthor >< id) .
  sort .
  map (swap >< sort) .
  collect .
  rinv .
  rcomp .
  (discollect >< discollect . getb)
)

```

Para as constantes disponibilizadas no anexo C, esta função deverá dar o resultado seguinte:

```

Main> mkInd bcp bib
[(Author{initials="M.A.",surname="Arbib"},[10]),

```

```
(Author{initials="L.S.",surname="Barbosa"},[28]),
(Author{initials="J.B.",surname="Barros"},[2,28,29]),
(Author{initials="E.",surname="Fielding"},[24,25]),
(Author{initials="P.H.B.",surname="Gardiner"},[8]),
(Author{initials="J.A.",surname="Goguen"},[28]),
(Author{initials="E.",surname="Horowitz"},[2,16]),
(Author{initials="C.B.",surname="Jones"},[3,7,28]),
(Author{initials="I.S.",surname="Jourdan"},[11,17,29]),
(Author{initials="J.",surname="Le Goff"},[10]),
(Author{initials="E.G.",surname="Manes"},[10]),
(Author{initials="C.",surname="Morgan"},[8]),
(Author{initials="F.S.",surname="Moura"},[28]),
(Author{initials="J.N.",surname="Oliveira"},[2,9,10,17,28,29]),
(Author{initials="C.J.",surname="Rodrigues"},[29]),
(Author{initials="S.",surname="Sahni"},[2,16]),
(Author{initials="J.M.",surname="Spivey"},[3,7]),
(Author{initials="J.M.",surname="Valenca"},[2,28,29]),
(Author{initials="N.",surname="Wirth"},[2,3]),
(Author{initials="J.C.P.",surname="Woodcock"},[29])) :: [(Author,[Int])]
Main>
```

4.2 Reconhecimento do formato BIB_TEX

No anexo B dispõe de funções para reconhecimento de (parte do) formato BIB_TEX. Faça

```
parse parseItem ex1
```

ou

```
parse parseBib exemplo
```

para testar, onde `ex1` e `exemplo` são strings (ver anexo C).

5 O que há a fazer

1. Realizar a função `mkInd` da forma sugerida (ou outra, desde que justificadamente melhor).
2. Testar o seu comportamento (pelo menos) nas situações seguintes:
 - (a) Para as constantes dadas no anexo C.
 - (b) Para a situação em que `bkp` cita chaves que não se encontram em `bib`.
 - (c) Para a situação em que `bib` contém mais entradas bibliográficas que as que ocorrem em `bkp`.
 - (d) Para a situação em que `bib` repete entradas de bibliografia.
3. “Colagem” ao L^AT_EX: realizar funções monádicas de leitura e escrita da informação nos ficheiros relevantes.
4. Compilar a versão final em GHC, por forma ficar disponível um filtro independente do HUGS, capaz de ser utilizado por qualquer adepto de L^AT_EX, em qualquer sistema operativo onde L^AT_EX esteja instalado.
5. Testar tudo com o presente documento (`mpi0102t2.lhs`)¹.

¹Esta nota de rodapé repete algumas citações bibliográficas para efeitos de teste: [1, 4].

5.1 Sugestões para Valorização

- Realizar funcionalidade para construção de histogramas de citação (por anos e por autores).
- Exportar índice de autores em formato HTML e CSV (para importação em folhas de cálculo, *etc.*)
- Estender o reconhecimento a todo o formato BIB_TEX.

A Funções por implementar

As seguintes funções são relevantes para a realização do seu trabalho. Delas apenas se dá a respectiva assinatura. Para que o HUGS as interprete sem erros todas foram reduzidas à função constante “lista vazia”. Esforce-se por defini-las em notação “pointfree”.

```
dunion :: Eq a => [[a]] -> [a]
dunion = const []

discollect :: (Eq a, Eq b) => [(b,[a])] -> [(b,a)]
discollect = const []

rinv :: [(a,b)] -> [(b,a)]
rinv = const []

sell :: Eq a => a -> [(a,b)] -> [(a,b)]
sell a = const []

sel2 :: Eq a => a -> [(b,a)] -> [(b,a)]
sel2 b = const []

rcomp :: Eq a => ([(b,a)],[(a,c)]) -> [(b,c)]
rcomp = const []

collect :: (Eq a, Eq b) => [(a,b)] -> [(a,[b])]
collect = const []

dom :: [(a,b)] -> [a]
dom = const []

rng :: Eq a => [(b,a)] -> [a]
rng = const []
```

B Sugestões para “parser” de formato BIB_TE_X

```
data BibItem = Item String String [(String,String)] deriving Show

parseBib :: Parse [BibItem]
parseBib = parseSeq parseItem

parseItem :: Parse BibItem
parseItem = do  parseConst "@" ;
                tipo <- parseString ;
                (autor, pares) <- parseChaveta parseAtribs ;
                return (Item tipo autor pares)

parseAtribs = do  autor <- parseStringTo ',' ;
                 parseConst "," ;
                 pares <- parseSeqSep "," parseAtrib ;
                 return (autor,pares)

parseAtrib :: Parse (String,String)
parseAtrib = do  x <- parseString ;
                 parseConst "=" ;
                 y <- parseString ;
                 return (x,y)
```

C Constantes para validação

```
bkp :: Bkp
bkp = [
    (2,["Wi76","HS??"]),
    (2,["VB00","Ol92"]),
    (3,["Wi76"]),
    (3,["Jo86"]),
    (3,["Sp89"]),
    (3,["Jo80"]),
    (7,["Jo86"]),
    (7,["Sp89"]),
    (8,["MG90"]),
    (9,["NG90"]),
    (9,["Ol92"]),
    (10,["Le84"]),
    (10,["Ol92"]),
    (10,["MA86"]),
    (11,["Jo92"]),
    (16,["HS??"]),
    (17,["Jo92","Ol92"]),
    (24,["Fi80"]),
    (25,["Fi80"]),
    (28,["Go86"]),
    (28,["OM95"]),
    (28,["Ba95"]),
    (28,["Ba95"]),
    (28,["Jo80"]),
    (28,["Ol92","VB00"]),
    (29,["Wo92"]),
    (29,["VB00"]),
    (29,["Ol90","Ol92","Jo92","Ro93"])
]
```



```

bib :: Bib
bib = [
    "Ba95" |-> [ Author "L.S." "Barbosa" ],
    "Fi80" |-> [ Author "E." "Fielding" ],
    "Le84" |-> [ Author "J." "Le Goff" ],
    "Go86" |-> [ Author "J.A." "Goguen" ],
    "HS??" |-> [ Author "E." "Horowitz" , Author "S." "Sahni" ],
    "Jo80" |-> [ Author "C.B." "Jones" ],
    "Jo86" |-> [ Author "C.B." "Jones" ],
    "Jo92" |-> [ Author "I.S." "Jourdan" ],
    "MA86" |-> [ Author "E.G." "Manes" , Author "M.A." "Arbib" ],
    "MG90" |-> [ Author "C." "Morgan" , Author "P.H.B." "Gardiner" ],
    "OC93" |-> [ Author "J.N." "Oliveira" , Author "A.M." "Cruz" ],
    "OM95" |-> [ Author "J.N." "Oliveira" , Author "F.S." "Moura" ],
    "Ol90" |-> [ Author "J.N." "Oliveira" ],
    "Ol92" |-> [ Author "J.N." "Oliveira" ],
    "Ro93" |-> [ Author "C.J." "Rodrigues" ],
    "Sp89" |-> [ Author "J.M." "Spivey" ],
    "VB00" |-> [ Author "J.M." "Valenca" , Author "J.B." "Barros" ],
    "Wi76" |-> [ Author "N." "Wirth" ],
    "Wo92" |-> [ Author "J.C.P." "Woodcock" ]
]

exemplo :: String
exemplo = ex1 ++ ex2

ex1 :: String
ex1 = concat [
    "@bookoprw1,",
    "  title          = Resolution of Equations in Algebraic Structures.",
    "                  Volume 2: Rewriting Techniques",
    "  editor          = Maurice Nivat and Hassan Ait-Kaci",
    "  year            = 1989,",
    "  publisher       = Academic Press, New York",
    "  key             = Nivat and Ait-Kaci" ]

ex2 :: String
ex2 = concat [
    "@inproceedingsGB:rta89,",
    "  author = Garrett Birkhoff",
    "  month = apr,",
    "  year = 1989,",
    "  title = Rewriting Ideas from Universal Algebra",
    "  booktitle = rta89,",
    "  editor = nd,",
    "  address = ch,",
    "  pages = i",
    "  note = lnCS355",
    "" ]

```

References

- [1] Simon Peyton Jones (ed.), John Hughes (ed.), Lennart Augustsson, Dave Barton, Brian Boutel, Warren Burton, Joseph Fasel, Kevin Hammond, Ralf Hinze, Paul Hudak, Thomas Johnsson, Mark Jones, John Launchbury, Erik

Meijer, John Peterson, Alastair Reid, Colin Runciman, and Philip Wadler. Report on the programming language Haskell 98 — a non-strict, purely functional language. Technical report, February 1999.

- [2] P. Hudak. *The Haskell School of Expression - Learning Functional Programming Through Multimedia*. Cambridge University Press, 1st edition, 2000. ISBN 0-521-64408-9.
- [3] L. Lamport. *LaTeX — A Document Preparation System*. Addison-Wesley Publishing Company, 5 edition, 1986.
- [4] J.M. Valença and J.B. Barros. *Fundamentos da Computação II: Programação funcional*. Universidade Aberta, 2000. ISBN 972-674-318-4, 234 p.

Índice de Autores

[vazio por omissão]