

Faculdade de Engenharia da Universidade do
Porto

Manipulação Simbólica de Polinómios em Haskell

Trabalho Prático 1

Programação Funcional e em Lógica
Turma 5 - Grupo 5

- Fábio Araújo de Sá (up202007658@fe.up.pt)
- Lourenço Alexandre Correia Gonçalves (up202004816@fe.up.pt)

Porto, 17 de Outubro de 2022

Índice

Representação Interna de Polinómios	3
Parser	4
Operações com Polinómios	5
Normalização	5
Adição	6
Multiplicação	6
Derivação	7
Testes	8

Representação Interna de Polinómios

Para realizar manipulação simbólica de polinómios em Haskell é muito importante que estes tenham uma representação interna que seja correta, simples e consistente, de modo aumentar a facilidade e eficiência da realização de operações sobre os mesmos.

Com isto em mente, decidimos utilizar a seguinte representação:

- Um polinómio é uma lista de monómios
- Um monómio é constituído por dois elementos, o coeficiente e uma parte literal
- O coeficiente é um número inteiro, a parte literal é uma lista de pares em que o primeiro elemento é a variável e o segundo é o seu expoente
- Uma variável é um carater e o expoente é um número inteiro não negativo
- Para termos independentes, a variável é um espaço ' ' e o seu expoente é 0

Assim, o polinómio:

$$2x^3 + 3x^2y - 4$$

É representado da seguinte forma:

```
[(2, [( 'x', 3 )]), (3, [( 'x', 2 ), ( 'y', 1 )]), (-4, [])]
```

Os tipos em Haskell foram definidos da forma que se segue:

```
type Coefficient = Int
type Exponent = Int
type Symbol = Char
type Monomial = (Coefficient, [(Symbol, Exponent)])
type Polynomial = [Monomial]
```

Parser

A função do parser é transformar um polinómio em formato de string num polinómio na nossa representação interna, explicada anteriormente. O formato em string melhora e simplifica a interação com o utilizador.

Um polinómio em formato de string considera-se válido se respeitar as seguintes regras:

1. As variáveis dos monómios devem estar separadas por um '*'
Ex: "xy" e "x^2y" são inválidas, devendo ser representadas como "x*y" e "x^2*y", respetivamente.
2. O coeficiente e a parte literal de um monómio podem ou não ser separados por asterisco ("-2x*y" e "-2*x*y" são equivalentes).

Exemplo:

Considere-se o input " $2x^3 + 3x^2y - 4$ ". O procedimento do parser é o seguinte:

filter (not.isSpace) \rightarrow " $2x^3+3x^2y-4$ "

(splitSignal) \rightarrow [" $2x^3$ ", " $+3x^2y$ ", " -4 "]

(removePlus) \rightarrow [" $2x^3$ ", " $3x^2y$ ", " -4 "]

(toMonomial) \rightarrow [(2, " x^3 "), (3, " x^2y "), (-4, "")]

(formatLitPart) \rightarrow [(2, [" x^3 "]), (3, [" x^2 ", "y"]), (-4, [])]

(formatVariables) \rightarrow [(2, [(('x', 3)]), (3, [(('x', 2), ('y', 1))], (-4, [])]

A indentação significa que a função é executada dentro da função da linha acima

A função polyToString transforma, tal como o nome indica, um polinómio na representação interna numa string. O procedimento é em tudo semelhante mas inverso ao indicado anteriormente.

Operações com Polinómios

Normalização

Considere-se o input “ $4 + 5x^*y^2 + x^4 - 2y^2*x - x^4$ ”, com a respetiva notação interna:

`[(4, []), (5, [(‘x’, 1), (‘y’, 2)]), (1, [(‘x’, 4)]), (-2, [(‘y’, 2), (‘x’, 1)]), (-1, [(‘x’, 4)])]`

1ª Etapa: sortExponents

Ordenação da parte literal de cada monómio alfabeticamente

`[(4, []), (5, [(‘x’, 1), (‘y’, 2)]), (1, [(‘x’, 4)]), (-2, [(‘x’, 1), (‘y’, 2)]), (-1, [(‘x’, 4)])]`

2ª Etapa: getUniqueTerms

Criação de um novo polinómio com todos os coeficientes iguais a 0 e todas as partes literais diferentes

`[(4, []), (0, [(‘x’, 1), (‘y’, 2)]), (0, [(‘x’, 4)])]`

3ª Etapa: doAllSums

Com auxílio das funções ‘foldl’ e ‘doAllSums’, acumula-se no polinómio resultante da etapa 2 os expoentes do polinómio inicial (onde as partes literais coincidem)

`[(4, []), (3, [(‘x’, 1), (‘y’, 2)]), (0, [(‘x’, 4)])]`

4ª Etapa: filter

Durante as operações, poderão existir monómios em que o coeficiente seja nulo. Estes devem ser retirados

`[(4, []), (3, [(‘x’, 1), (‘y’, 2)])]`

5ª Etapa: sortBy (\x y -> compare (getDegree y) (getDegree x))

Finalmente, o polinómio é ordenado segundo o grau dos seus monómios

`[(3, [(‘x’, 1), (‘y’, 2)]), (4, [])]`

Adição

Considere-se os inputs “ $4 + 5x^2y + x^4$ ” e “ $-3y^2x - x^5$ ”, com as respetivas notações internas:

$[(4, []), (5, [(x', 1), (y', 2)]), (1, [(x', 4)])]$ e $[(-3, [(y', 2), (x', 1)]), (-1, [(x', 5)])]$

1ª Etapa: (++)

Concatenação das duas listas de monómios

$[(4, []), (5, [(x', 1), (y', 2)]), (1, [(x', 4)]), (-3, [(y', 2), (x', 1)]), (-1, [(x', 5)])]$

2ª Etapa: normalize

O polinómio resultante é normalizado seguindo as etapas da operação normalize

$[(-1, [(x', 5)]), (1, [(x', 4)]), (3, [(x', 1), (y', 2)]), (4, [])]$

Multiplicação

Considere-se os inputs “ $4 + 5x$ ” e “ $-3x^2z^2$ ”, com as respetivas notações internas:

$[(4, []), (5, [(x', 1)])]$ e $[(-3, [(x', 2), (z', 2)])]$

1ª Etapa: multiMonomial

Consideram-se todas as combinações possíveis entre monómios de polinómios diferentes. Para cada combinação multiplicam-se os coeficientes dos monómios e a multiplicação dos expoentes fica a cargo das funções multiExponents e getUniqueSymbols

$[(-12, \text{multiExponents } [] [(x', 2), (z', 2)]), (-15, \text{multiExponents } [(x', 1)] [(x', 2), (z', 2)])]$

2ª Etapa: getUniqueSymbols

A técnica é semelhante à usada na função getUniqueTerms da operação de normalização. Criação de uma nova lista de termos com todos os expoentes nulos e todas as partes literais diferentes.

$[(x', 0), (z', 0)]$ e $[(x', 0), (z', 0)]$

3ª Etapa: multiExponents

A técnica é semelhante à usada na função doAllSums da operação de normalização. Com auxílio da função 'foldl', acumula-se na lista resultante da etapa 2 os expoentes do polinómio inicial (onde as variáveis coincidem)

$[(-12, [(x', 2), (z', 2)]), (-15, [(x', 3), (z', 2)])]$

4ª Etapa: normalize

O polinómio resultante é normalizado seguindo as etapas da operação normalize.

$[(-15, [(x', 3), (z', 2)]), (-12, [(x', 2), (z', 2)])]$

Derivação

Considere-se os inputs 'x' e " $4 + 5x*y^2 + x^4 + z$ ", tendo o polinómio a respetiva notação interna:

$[(4, []), (5, [(x', 1), (y', 2)]), (1, [(x', 4)]), (1, [(z', 1)])]$

1ª Etapa: deriveMonomial

Para monómios que representem um termo independente, o seu coeficiente é colocado a 0. Nos outros casos, percorre-se toda a parte literal (pares variável, expoente). Se nenhuma variável coincidir com o caracter de input, então o coeficiente é colocado a 0. Caso contrário:

- o novo coeficiente do monómio é a multiplicação do expoente respetivo e o anterior coeficiente
- o expoente da variável é subtraído de uma unidade

$[(0, []), (5, [(x', 0), (y', 2)]), (4, [(x', 3)]), (0, [(z', 1)])]$

2ª Etapa: normalize

O polinómio resultante é normalizado seguindo as etapas da operação normalize.

$[(4, [(x', 3)]), (5, [(y', 2)])]$

Testes

Para verificar a correção do nosso código foram feitos testes unitários, testando operações simples (*normalize*, *add*, *multi*, *derive*), assim como algumas das suas propriedades algébricas.