TP1_G10_04

Representação interna

Para o nosso projeto achamos melhor guardar um polinómio como uma lista de monômios e este organizamos num tuplo de 3 elementos, em que o primeiro é um inteiro que representa o coeficiente e o segundo e terceiro elementos são listas, a primeira de caracteres para a parte literal e a segunda de inteiros para o grau de cada variável, nestas listas a variável e o grau correspondente estão guardados na mesma posição. Este modelo de organização pareceu-nos o mais indicado porque apresenta maior facilidade de percorrer a informação através das listas e também tem a capacidade de usar funções como o zipWith para percorrer o grau e a variável, além de poder comparar as listas diretamente entre diferentes monómios.

Funcionalidades

Parsing

Ao receber o input, divide a string por monómios, e a cada monómio dá parse do coeficiente e depois das variáveis e graus recursivamente, dependendo do formato de cada monómio, por exemplo: o sinal do coeficiente, número de casas do coeficiente, número de variáveis, variáveis não existentes, expoente igual ou maior que 1, etc.

Normalizar

Para normalizar percorremos o polinómio e por cada monómio, juntámos a esse monómio os monómios com lista de variáveis e expoentes iguais e apagámos os mesmos. Por causa de casos como 2xyz + 2zxy foi preciso fazer uma função que ordena as variáveis dos monómios e com isso podémos simplesmente comparar listas de variáveis e listas de expoentes.

Adicionar

Para adicionar apenas demos append dos dois polinómios e normalizámos.

Multiplicar

Para multiplicar, transformámos ambos os polinómios numa lista de pares de monómios através dum gerador, ficando por exemplo "12" "34" = [('1','3'),('1','4'),('2','3'),('2','4')]. Depois multiplicaram-se ambos os monómios de cada par. Para fazer isso multiplcámos coeficientes e demos append das listas de variáveis de ambos e das listas de expoentes, ficando uma lista com variáveis repetidas (1, [x,x], [1,2]). Após isso fizemos um zip da lista de variáveis com lista de expoentes (1, [(x,1), (x,2)]), e por cada par de variável/expoente, somámos todos os elementos da lista com variável igual e apagámos os mesmos (1, [(x,3)]). Voltámos a converter para a estrutura inicial.

Derivar

Para aplicar as regras de derivação nós percorremos todos os monômios e em cada multiplicamos o coeficiente pelo o grau da derivada a derivar e decrementamos esse grau por um valor. Se o grau da derivada a derivar for zero, o monômio é removido da lista do polinómio.

Ordenar

Fizemos várias opções de ordenação de um polinómio e escolhemos para output ordem descendente por expoente máximo, seguida por ordem descendente de número de variáveis, de seguida por ordem alfabética de variáveis.

Testar

Numa primeira fase, podem ser corridos testes unitários, chamando as funções do ficheiro **Test.hs**. Já para correr o programa principal é preciso chamar a função main, esta vai pedir um input para escolher que tipo de operação fazer e de seguida vai pedir polinómios conforme a operação pertendida.

TODO:

- [x] Output ser string
- [x] Organizar
- [x] Normalizar um polinomio
- [x] Adicionar polinomios
- [x] Multiplicar polinomios
- [x] Derivado de um polinomio
- [x] Input ser string
- [ ] QuickCheck

G10_04

- [x] Carolina Brandão
- [x] Diogo Rodrigues
- [x] Jorge Costa