

Laboratórios de Informática I

2020/2021

Mestrado Integrado em Engenharia Informática



Universidade do Minho

Sessão Laboratorial 2



Introdução

- O executável da semana passada avalia uma função particular do programa fornecido – a função `main`
- Outras funções incluídas no programa só serão executadas se forem requeridas/usadas na avaliação da função `main`
- Comportamento não é o mais apropriado numa fase de desenvolvimento
- Interessa testar cada função separadamente

O Interpretador ghci

- O interpretador não produz qualquer ficheiro executável a partir de um programa *Haskell*
- Disponibiliza ao programador um ambiente onde pode avaliar qualquer expressão *Haskell*
- O interpretador é invocado pelo comando `ghci`
- Uma vez invocado, surge o *prompt* `Prelude>`
- Pode-se avaliar uma qualquer expressão *Haskell* (e.g. `3+2*2`)
- Ou um comando específico do interpretador (e.g. `:load Fich.hs`, que carrega o ficheiro tornando disponíveis as várias funções aí definidas)

Comandos

- `:?` ou `:help` — mostra informação sobre comandos do ghci
- `:quit` — sai do interpretador
- `:cd <dir>` — altera directoria corrente para `<dir>`
- `:load <mod>` — carrega módulo `<mod>` (ficheiro)
- `:reload` — recarrega último módulo
- `:type <expr>` — imprime tipo da expressão `<expr>`
- `:info <symb>` — imprime informação sobre símbolo `<symb>`
- `:!<cmd>` — invoca o comando Unix `<cmd>`

O ghci aceita abreviaturas dos comandos – e.g., o comando `:load HelloWorld` pode ser escrito `:l HelloWorld`.

Utilização de Bibliotecas

- O *Haskell* disponibiliza um conjunto de *bibliotecas* que oferecem mais funcionalidade
- Para utilizar uma biblioteca é necessário *importar* o respectivo módulo
- Por exemplo, no módulo `Data.Char` encontramos funções para manipular valores do tipo `Char` (*caracteres*)
- É necessário incluir a declaração `import Data.Char` no início do programa

Exemplos de Bibliotecas

- `Prelude` — conjunto de tipos e funções pré-carregados (i.e. não é necessário importar explicitamente qualquer módulo)
- `Data.Char` — funções de manipulação de caracteres
- `Data.String`: funções de manipulação de *strings* (i.e. sequências de caracteres)
- `Data.List` — funções de manipulação de listas
- `Data.Maybe` — funções para manipulação do tipo `Maybe`

Documentação

- Um recurso útil quando recorremos às bibliotecas é a sua documentação
- Aí encontramos qual a funcionalidade oferecida (quais os módulos; tipos e funções disponibilizados), assim como uma descrição sumária de cada função (incluindo o seu tipo)
- A documentação da biblioteca *standard* na página de documentação do ghc:
<http://www.haskell.org/ghc/docs/latest/html/>)
- No item *Libraries* encontrará todas as bibliotecas instaladas pelo GHC

Outros Apontadores

- <http://www.haskell.org>: página oficial da linguagem, que inclui apontadores para todo o tipo de documentação sobre a linguagem (em particular, a própria a especificação da linguagem)
- <http://www.haskell.org/hoogle/>: disponibiliza um mecanismo de busca sobre a documentação das bibliotecas
- <http://hackage.haskell.org>: sítio que agrega contribuições (*packages*) desenvolvidas em *Haskell*
- <https://www.fpcomplete.com/haskell/learn/>: tutoriais *online* da linguagem
- https://www.tutorialspoint.com/compile_haskell_online.php: ambiente *online* para compilar/executar programas *Haskell*