

207

AVALIAÇÃO LAZY E MÔNADAS PARA ENTRADA/SAÍDA NA LINGUAGEM HASKELL. *Juliana Kaizer Vizzotto, Antônio Carlos da R. Costa, Graçaliz P. Dimuro* (NAPI, Escola de Informática – UCPel)

Na linguagem de programação funcional Haskell, podemos escrever programas usando um estilo diferente para avaliar as expressões, que chamamos de avaliação Lazy. Na avaliação lazy, como o próprio nome diz, acontece uma avaliação preguiçosa, ou seja, um argumento de uma função somente é avaliado se o valor do argumento é necessário para computar o resultado global. Uma consequência da avaliação lazy é que ela permite que a linguagem manipule estruturas infinitas. Para avaliar completamente estas estruturas precisamos de um tempo infinito mas, com a avaliação lazy, é possível examinar partes finitas dessas estruturas de dados. Na linguagem Haskell as funções podem possuir definições recursivas e qualquer tipo recursivo pode conter objetos infinitos. Em relação à isso, a avaliação lazy, permite otimizar a execução dos programas evitando qualquer computação desnecessária. Para expressar computações interativas e aplicações onde entradas e saídas são características importantes, a linguagem Haskell utiliza uma estrutura matemática chamada mônada, para organizar as entradas e saídas. Este trabalho tem como objetivo salientar a importância desta característica de execução da linguagem Haskell. Para isso, será feita uma comparação de scripts com avaliação lazy e scripts sem esta avaliação. Também serão apresentados exemplos de como as mônadas são utilizadas para programas interativos.(FAPERGS/UCPel).