Centro de Informática - UFPE
Segundo Exercício Escolar — IF686 (2017.1)
Data: 22/06/2017 Horário: 8h-10h
Docente: Márcio Lopes Cornélio
Discente:

1. Leia a descrição abaixo e, em seguida, faça o que se pede.

   Uma fábrica de lâmpadas utiliza máquinas de fabricação distintas que produzem bulbos, soquetes e embalagens que são colocados em caixas separadas. Após a a produção destes elementos, duas máquinas produzem lâmpadas, juntando bulbos e soquetes e colocando cada lâmpada em uma embalagem. Cada lâmpada embalada é colocada em uma única caixa até que seja completamente preenchida com 50 lâmpadas. Quando preenchida, a caixa é transportada para um depósito, sendo substituída por uma caixa vazia.

(2,5) (a) Implemente, em Haskell, o que foi descrito acima, utilizando *mutable variables* (MVar). envolvendo tipos primitivos atômicos).

(2,5) 2. Implemente em Java as classes *Produtor*, *Consumidor* e *ProdutorConsumidor*. Esta última possui o método *main()*. O dados produzidos são valores do tipo inteiro. Não se pode utilizar classes da API de Java como, por exemplo, interface *BlockingQueue*, que possui métodos *put* e *take*, utlizando a implementação *ArrayBlockingQueue*.

(1,5) 3. Modifique a questão anterior para utilizar tipos primitivo atômicos ou a interface *BlockingQueue*.

4. Defina um tipo chamado conta *Conta* constituído de uma variável transacional do tipo inteiro. Defina as seguintes funções em Haskell:

(0,5) (a) $saque :: Conta -> Int -> STM()$ que realiza retirada de uma quantia de uma conta

(1,0) (b) $deposito :: Conta -> Int -> STM()$ que realiza um depósito em uma conta. Utilize a função *saque* para definir *deposito*

(1,0) (c) *saque*2, uma modificação da função *saque*, que bloqueia, caso o saldo da conta vá se tornar negativo.

(1,0) (d) Suponha que você possa retirar dinheiro de uma conta *A* se esta tiver saldo suficiente, senão, retira de uma conta *B*. Defina a função *saque*3, utilizando a função *orElse* e a função *saque*2.