

## Coq (3) - Questões

Crie um ficheiro Coq para desenvolver as provas das propriedades abaixo indicadas. Pode usar táticas automáticas.

1. Apresente uma especificação forte adequada a cada uma das funções abaixo indicadas. Faça a prova desses teoremas (especificação) e, por fim, proceda à extracção do respectivo programa Haskell.

- (a) Uma função que, dado um número natural  $n$  e um valor  $x$ , constroi uma lista de tamanho  $n$  cujos elementos são todos iguais a  $x$ .
- (b) Uma função que recebe uma lista de pares de números naturais, e produz a lista com as somas das partes constituintes de cada par da lista.

2. Recorde a função `count` que conta o número de ocorrências de um inteiro numa lista de inteiros.

```
Fixpoint count (z:Z) (l:list Z) {struct l} : nat :=  
  match l with  
  | nil => 0%nat  
  | (z' :: l') => if (Z.eq_dec z z')  
                  then S (count z l')  
                  else count z l'  
  end.
```

- (a) Prove a seguinte propriedade

`forall (x:Z) (a:Z) (l:list Z), x <> a -> count x (a :: l) = count x l`

- (b) Defina uma relação indutiva que descreva a relação entre o input e o output para a função `count`, ou seja, a sua especificação.
- (c) Prove que a função dada acima satisfaz a especificação apresentada na alínea anterior.