DESENVOLVIMENTO GUIADO POR TIPOS UFABC Q2.2022
SPECIFICATION PATTERN NO NIVEL DOS TIPOS

INTRODUÇÃO

Antes de tudo acredito que eu deva explicar o que é o specification pattern, de maneira simplória é uma forma de construir validadores e filtros dentro do seu projeto evitando repetições (reescrita) de condicionais, além disso esse pattern constrói uma forma de compor esses validadores criando condicionais cada vez mais complexos como faríamos com os operadores (And, Or e Not) nos tipos booleanos.
Essa solução é construída sempre no nível do termos (valores) sem que o compilador ou interpretador da sua linguagem preferida possa dizer que aquele tipo passou por validação X (e/ou) Y, dessa forma surgiu a motivação de implementar este padrão no nível dos tipos e ter em tempo de compilação a garantia de que um filtro em uma lista gerou uma coleção de objetos com as especificações requeridas ou que o parâmetro passado para uma função tenha as devidas especificações.

IMPLEMENTAÇÃO

A implementação tem como ideia simular um software de contratação de espiões por habilidades e permissões fornecidas no nível dos tipos, essa base de permissões possibilita criar especificações mais complexas utilizando os operadores lógicos também construídos no nível dos tipos. Um exemplo que existe na aplicação e ilustra bem o que foi dito acima é a criação de uma missão que ocorre em agua, ela precisa que um determinado espião tenha permissão para realizá-la e essa permissão é composta por permissões mais básicas relacionadas aos veículos que os espiões podem dirigir.

A segurança no nível dos tipos esta relacionada a criação de novas estruturas na aplicação como o exemplo da missão em água, isso garante consistência no domínio do seu software, além disso o pattern consegue ser utilizado como filtro de coleções garantindo que a coleção de itens de saída satisfaça todas as especificações impostas.

DIFICULDADES

A parte mais complexa no desenvolvimento foi ter a ideia da arvore de expressões no nivel dos tipos, após isso a implementação do pattern foi mais tranquila. Vale ressaltar que não tive o comportamento esperado no operador Not, um dos problemas é que ele não me garante que a entidade X não pode fazer a atividade Y, basta não avaliar a entidade para aquela atividade e o compilador ficara feliz, pensei em contornar isso com type families criando link entre o tipo "Poder fazer Y" e "Não pode fazer Y", entretanto teria que rever algumas outras type families.

INSTRUÇÕES

Para execução do programa você pode utilizar stack run, entretanto o código disponibilizado em app/Main.hs é mais útil para leitura e ver como o código funciona.
O vídeo com uma breve explicação do projeto esta disponível em: https://youtu.be/31V1tfA4Ys8

Guilherme Rocha Muzi Franco 11201920623