Trabalho final de Haskell

O objetivo desse trabalho é implementar um algoritmo de unificação para uma versão bastante simplificada do sistema de tipos de Haskell. O trabalho pode ser feito em duplas.

Especificação

Considere o seguinte código:

```
import Text.ParserCombinators.Parsec

data Type = TypeInt
          | TypeVar Name
          | TypeArrow Type Type
          deriving Show

type Name = String

type Unifier = [(Name, Type)]
```

O programa deve então solicitar ao usuário dois tipos e informar se tais tipos podem ser unificados, informando qual o unificador mais geral, ou se a unificação não pode ser feita. Para tal, recomenda-se a utilização da biblioteca Parsec para a implementação de um simples parser para interpretar os tipos de entrada.

A estrutura principal do programa se dá através da função main:

```
main :: IO ()
```

A função principal para unificação receberá dois tipos e tentará retornar um unificador mais geral na forma de `Just mgu`, retornando `Nothing` caso não seja possível unificar.

```
unify :: Type -> Type -> Maybe Unifier
```

Para sua implementação, duas funções auxiliares são necessárias, que respectivamente verificam se uma variável aparece livre em um tipo, e que compõe duas unificações distintas.

```
occursCheck :: Name -> Type -> Bool
compose :: Unifier -> Unifier -> Unifier
```

A fim de se testar o sistema, uma função deve ser implementada capaz de aplicar uma substituição a um tipo arbitrário, retornando um novo tipo.

```
subst :: Unifier -> Type -> Type
```

Considere também as seguintes funções para o reconhecimento de texto, definindo a seguinte gramática para tipos:

```
parseType :: Parser Type     -- type: function | atom
parseAtom :: Parser Type     -- atom: int | var | paren
parseInt :: Parser Type      -- int: "Int"
parseVar :: Parser Type      -- var: lowercase+
parseFun :: Parser Type      -- fun: atom "->" type
parseParen :: Parser Type    -- paren: "(" type ")"
```

Isto é, conforme visto em sala, temos um tipo primitivo Int, variáveis de tipo, e tipos de funções representados pelas setas. O tipo das funções pode variar caso o aluno opte por não usar a biblioteca Parsec.

Entrega do trabalho

A combinar.