

# Codificări Church

## Laborator 12

În acest laborator, vom face experimente cu codificări Church în lambda-calcul.

**Exercițiul 0.1.** Funcțiile `reduce1` și `reduce1'` vor fi utilizate pentru testare:

```
data Term = Var Id
          | App Term Term
          | Lambda Id Term deriving (Show, Eq)
reduce1' :: Term -> [Id] -> Maybe Term
reduce1' (Var id') _ = Nothing
reduce1' (App (Lambda id term) term') avoid =
  Just (casubst id term' term avoid)
reduce1' (App term1 term2) avoid = case reduce1' term1 avoid of
  Nothing -> case reduce1' term2 avoid of
    Nothing -> Nothing
    Just term2' -> Just (App term1 term2')
  Just term1' -> Just (App term1' term2)
reduce1' (Lambda id term) avoid = case reduce1' term avoid of
  Nothing -> Nothing
  Just term' -> Just (Lambda id term')
reduce1 :: Term -> Maybe Term
reduce1 t = reduce1' t (vars t)
```

Testați funcțiile de mai sus pe câteva exemple.

**Notă:** Mai departe, pentru testare veți avea nevoie și de funcția `reduce` din laboratorul precedent.

**Exercițiul 0.2.** Implementați și testați codificările Church discutate la curs pentru:

1. Valori și operații booleene: `true`, `false`, `and`, `or`, `not`;
2. Numere naturale și operații peste numere naturale: `succ`, `plus`, `mult`;
3. Predicate: `isZero`, `≤`, `=`;
4. Perechi de numere naturale (`pair`) și operațiile: `fst`, `snd`;
5. Liste de numere naturale (`nil`, `cons`) și operațiile: `head`, `tail`, `isnil`