Taller de Inferencia de Tipos
Machete

Paradigmas de Lenguajes de Programación

1 Algoritmo de inferencia

- $\mathbb{W}(x) \stackrel{\text{def}}{=} \{x : s\} \triangleright x : s, \quad s$ variable fresca
- $\mathbb{W}(0) \stackrel{\text{def}}{=} \emptyset \triangleright 0 : nat$
- $\mathbb{W}(true) \stackrel{\text{def}}{=} \emptyset \triangleright true : bool$
- $\mathbb{W}(false) \stackrel{\text{def}}{=} \emptyset \triangleright false : bool$
- $\mathbb{W}(succ(U)) \stackrel{\text{def}}{=} S\Gamma \triangleright S\, succ(M) : nat$ donde
  - $\mathbb{W}(U) = \Gamma \triangleright M : \tau$
  - $S = MGU\{\tau \doteq nat\}$
- $\mathbb{W}(pred(U)) \stackrel{\text{def}}{=} S\Gamma \triangleright S\, pred(M) : nat$ donde
  - $\mathbb{W}(U) = \Gamma \triangleright M : \tau$
  - $S = MGU\{\tau \doteq nat\}$
- $\mathbb{W}(iszero(U)) \stackrel{\text{def}}{=} S\Gamma \triangleright S\, iszero(M) : bool$ donde
  - $\mathbb{W}(U) = \Gamma \triangleright M : \tau$
  - $S = MGU\{\tau \doteq nat\}$
- $\mathbb{W}(if\ U\ then\ V\ else\ W) \stackrel{\text{def}}{=} S\Gamma_1 \cup S\Gamma_2 \cup S\Gamma_3 \triangleright S(if\ M\ then\ P\ else\ Q) : S\sigma$ donde
  - $\mathbb{W}(U) = \Gamma_1 \triangleright M : \rho$
  - $\mathbb{W}(V) = \Gamma_2 \triangleright P : \sigma$
  - $\mathbb{W}(W) = \Gamma_3 \triangleright Q : \tau$
  - $S = MGU\{\sigma \doteq \tau, \rho \doteq bool\} \cup \{\sigma_1 \doteq \sigma_2 \mid x : \sigma_1 \in \Gamma_i, x : \sigma_2 \in \Gamma_j, i \neq j\}$
- $\mathbb{W}(\lambda x.U) \stackrel{\text{def}}{=} \Gamma' \triangleright \lambda x : \tau'.M : \tau' \to \rho$ donde
  - $\mathbb{W}(U) = \Gamma \triangleright M : \rho$
  - $\tau' = \begin{cases} \alpha \text{ si } x : \alpha \in \Gamma \\ s \text{ con } s \text{ variable fresca en otro caso} \end{cases}$
  - $\Gamma' = \Gamma \ominus \{x\}$
- $\mathbb{W}(U\,V) \stackrel{\text{def}}{=} S\Gamma_1 \cup S\Gamma_2 \triangleright S(M\,N) : St$ donde
  - $\mathbb{W}(U) = \Gamma_1 \triangleright M : \tau$
  - $\mathbb{W}(V) = \Gamma_2 \triangleright N : \rho$
  - t variable fresca
  - $S = MGU\{\tau \doteq \rho \to t\} \cup \{\sigma_1 \doteq \sigma_2 \mid x : \sigma_1 \in \Gamma_1, x : \sigma_2 \in \Gamma_2\}$

2 Código auxiliar

2.1 Expresiones

```
data Exp a = VarExp Symbol |
             ZeroExp |
             SuccExp (Exp a)|
             PredExp (Exp a) |
             IsZeroExp (Exp a) |
             TrueExp |
             FalseExp |
             IfExp (Exp a) (Exp a) (Exp a) |
             LamExp Symbol a (Exp a) |
             AppExp (Exp a) (Exp a)

type Symbol = String
type PlainExp = Exp ()
type AnnotExp = Exp Type
```

2.2 Tipos

```
data Type = TVar Int | TNat | TBool | TFun Type Type
```

2.3 Contexto

```
emptyEnv :: Env
extendE :: Env -> Symbol -> Type -> Env
removeE :: Env -> Symbol-> Env
evalE :: Env -> Symbol -> Type
joinE :: [Env] -> Env
domainE :: Env -> [Symbol]
```

2.4 Sustituciones

```
emptySubst :: Subst
extendS :: Int -> Type -> Subst -> Subst

class Substitutable a where
    (<.>) :: Subst -> a -> a
    instance Substitutable Type   -- subst <.> t
    instance Substitutable Env    -- subst <.> env
    instance Substitutable Exp    -- subst <.> e
```

2.5 Unificación

```
type UnifGoal = (Type, Type)
data UnifResult = UOK Subst | UError Type Type
mgu :: [UnifGoal] -> UnifResult
```