

TDA *Bool*

Specificare

· Constructori: $\left| \begin{array}{l} T : \rightarrow Bool \\ F : \rightarrow Bool \end{array} \right.$

· Operatori: $\left| \begin{array}{l} not : Bool \rightarrow Bool \\ and : Bool^2 \rightarrow Bool \\ or : Bool^2 \rightarrow Natural \end{array} \right.$

· Axiome: $\left| \begin{array}{l} not : not(T) = F \\ \quad \quad not(F) = T \\ and : and(T, a) = a \\ \quad \quad and(F, a) = F \\ or : or(T, a) = T \\ \quad \quad or(F, a) = a \end{array} \right.$

- Intuiție: **selecția** între cele două valori, *true* și *false*
- $T \equiv_{\text{def}} \lambda x y.x$
- $F \equiv_{\text{def}} \lambda x y.y$
- Comportament de **selector**:
 - $(T a b) \rightarrow (\lambda x y.x a b) \rightarrow a$
 - $(F a b) \rightarrow (\lambda x y.y a b) \rightarrow b$

- $not \equiv_{\text{def}} \lambda x.(x F T)$
 - $(not T) \rightarrow (\lambda x.(x F T) T) \rightarrow (T F T) \rightarrow F$
 - $(not F) \rightarrow (\lambda x.(x F T) F) \rightarrow (F F T) \rightarrow T$

- $and \equiv_{\text{def}} \lambda x y.(x y F)$
 - $(and T a) \rightarrow (\lambda x y.(x y F) T a) \rightarrow (T a F) \rightarrow a$
 - $(and F a) \rightarrow (\lambda x y.(x y F) F a) \rightarrow (F a F) \rightarrow F$

- $or \equiv_{\text{def}} \lambda x y.(x T y)$
 - $(or T a) \rightarrow (\lambda x y.(x T y) T a) \rightarrow (T T a) \rightarrow T$
 - $(or F a) \rightarrow (\lambda x y.(x T y) F a) \rightarrow (F T a) \rightarrow a$

- Axiome:

- $(if\ T\ a\ b) \rightarrow a$
- $(if\ F\ a\ b) \rightarrow b$

- Implementare: $if \equiv_{\text{def}} \lambda c t e.(c t e)$

- $(if\ T\ a\ b) \rightarrow (\lambda c t e.(c t e)\ T\ a\ b) \rightarrow (T\ a\ b) \rightarrow a$
- $(if\ F\ a\ b) \rightarrow (\lambda c t e.(c t e)\ F\ a\ b) \rightarrow (F\ a\ b) \rightarrow b$

- Funcție **nestructă!**

- Constructori de bază:

- $pair : A \times B \rightarrow Pair$

- Operatori:

- $fst : Pair \rightarrow A$

- $snd : Pair \rightarrow B$

- Axiome:

- $fst(pair(a, b)) = a$

- $snd(pair(a, b)) = b$

- Intuiție: pereche \rightarrow funcție ce așteaptă **selectorul**, pentru a-l aplica asupra membrilor
- $pair \equiv_{\text{def}} \lambda x y z. (z x y)$
 - $(pair a b) \rightarrow (\lambda x y z. (z x y) a b) \rightarrow \lambda z. (z a b)$
- $fst \equiv_{\text{def}} \lambda p. (p T)$
 - $(fst (pair a b)) \rightarrow (\lambda p. (p T) \lambda z. (z a b)) \rightarrow (\lambda z. (z a b) T) \rightarrow (T a b) \rightarrow a$
- $snd \equiv_{\text{def}} \lambda p. (p F)$
 - $(snd (pair a b)) \rightarrow (\lambda p. (p F) \lambda z. (z a b)) \rightarrow (\lambda z. (z a b) F) \rightarrow (F a b) \rightarrow b$

· Constructori: $\left\{ \begin{array}{l} nil : \rightarrow List \\ cons : A \times List \rightarrow List \end{array} \right.$

· Operatori: $\left\{ \begin{array}{l} car : List \setminus \{nil\} \rightarrow A \\ cdr : List \setminus \{nil\} \rightarrow List \\ null : List \rightarrow Bool \\ append : List^2 \rightarrow List \end{array} \right.$

· Axiome:

$car : car(cons(e, L)) = e$

$cdr : cdr(cons(e, L)) = L$

$null : null(nil) = T$

$null(cons(e, L)) = F$

$append : append(nil, B) = B$

$append(cons(e, A), B) = cons(e, append(A, B))$

TDA List

Implementare

- Intuiție: listă \rightarrow pereche (*head*, *tail*)
- $nil \equiv_{\text{def}} \lambda x. T$
- $cons \equiv_{\text{def}} pair$
 - $(cons\ e\ L) \rightarrow (\lambda x\ y\ z.(z\ x\ y)\ e\ L) \rightarrow \lambda z.(z\ e\ L)$
- $car \equiv_{\text{def}} fst$
- $cdr \equiv_{\text{def}} snd$
- $null \equiv_{\text{def}} \lambda L.(L\ \lambda x\ y.F)$
 - $(null\ nil) \rightarrow (\lambda L.(L\ \lambda x\ y.F)\ \lambda x.T) \rightarrow (\lambda x.T\ \dots) \rightarrow T$
 - $(null\ (cons\ e\ L)) \rightarrow (\lambda L.(L\ \lambda x\ y.F)\ \lambda z.(z\ e\ L)) \rightarrow (\lambda z.(z\ e\ L)\ \lambda x\ y.F) \rightarrow (\lambda x\ y.F\ e\ L) \rightarrow F$

• *append* \equiv_{def}

$\lambda A B. (\text{if } (\text{null } A) B (\text{cons } (\text{car } A) (\text{append } (\text{cdr } A) B)))$

· Problemă: expresia **nu** admite formă închisă! → vezi eliminarea recursivității textuale.

- Intuiție: număr \rightarrow **listă** cu lungimea egală cu valoarea numărului
- $zero \equiv_{\text{def}} nil$
- $succ \equiv_{\text{def}} \lambda n. (cons\ nil\ n)$
- $pred \equiv_{\text{def}} cdr$
- $zero? \equiv_{\text{def}} null$
- $add \equiv_{\text{def}} append$