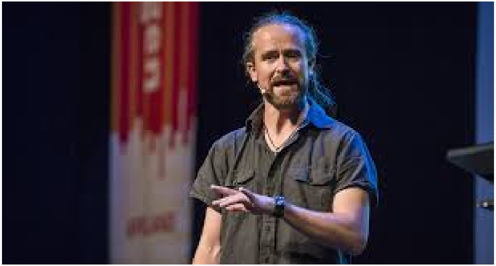
Programare funcțională

Introducere în programarea funcțională folosind Haskell
Corespondența Curry-Howard (material opțional)

Claudia Chiriță
Denisa Diaconescu

Departamentul de Informatică, FMI, UB

Schimbați perspectiva



Roger Antonsen

Universitatea din Oslo

TED Talk: Math is the hidden secret to understanding the world

"... înțelegerea constă în abilitatea de a-ți schimba perspectiva"

https://www.ted.com/talks/roger_antonsen_math_is_the_hidden_secret_to_understanding_the_world

Un program simplu în Haskell

```haskell
data Point = Point Int Int

makePoint :: Int -> Int -> Point
makePoint x y = Point x y

getX :: Point -> Int
getX (Point x y) = x

getY :: Point -> Int
getY (Point x y) = y

origin :: Point
origin = makePoint 0 0
```

Un program simplu în Haskell

Hai să schimbăm perspectiva!

```haskell
data Point = Point Int Int

makePoint :: Int -> Int -> Point
makePoint x y = Point x y
```

$$\frac{x : Int \quad y : Int}{makePoint\ x\ y : Point}\ (Point_I)$$

```haskell
getX :: Point -> Int
getX (Point x y) = x
```

$$\frac{p : Point}{getX\ p : Int}\ (Point_{E_1})$$

```haskell
getY :: Point -> Int
getY (Point x y) = y
```

$$\frac{p : Point}{getY\ p : Int}\ (Point_{E_2})$$

Generalizare

$$\frac{x : Int \quad y : Int}{makePoint\ x\ y : Point}\ (Point_I) \qquad \frac{M : A \quad N : B}{\langle M, N\rangle : A \times B}\ (\times_I)$$

$$\frac{p : Point}{getX\ p : Int}\ (Point_{E_1}) \qquad \frac{M : A \times B}{fst\ M : A}\ (\times_{E_1})$$

$$\frac{p : Point}{getY\ p : Int}\ (Point_{E_2}) \qquad \frac{M : A \times B}{snd\ M : B}\ (\times_{E_2})$$

Alt exemplu simplu

```
f = (\x -> x * 3) :: Int -> Int
```

$$\frac{\{x : Int\} \vdash x * 3 : Int}{\lambda x.x * 3 : Int \to Int}\ (fun_I)$$

```
> f 5
15
```

$$\frac{f : Int \to Int \quad 5 : Int}{f\ 5 : Int}\ (fun_E)$$

Generalizare

$$\frac{\{x : Int\} \vdash x * 3 : Int}{\lambda x.x * 3 : Int \to Int}\ (fun_I) \qquad \frac{\{x : A\} \vdash M : B}{\lambda x.M : A \to B}\ (\to_I)$$

$$\frac{f : Int \to Int \quad 5 : Int}{f\ 5 : Int}\ (fun_E) \qquad \frac{M : A \to B \quad N : A}{M\,N : B}\ (\to_E)$$

Logica. Ce este adevăt și ce este fals?

Hai să schimbăm perspectiva iar!

Logica. Ce este adevăt și ce este fals?

Hai să schimbăm perspectiva iar!

Dacă afară este întuneric atunci,
dacă porcii zboară atunci este întuneric afară.

$$\begin{aligned} A &= \text{afară este întuneric} \\ B &= \text{porcii zboară} \end{aligned} \qquad A \supset (B \supset A)$$

Logica. Ce este adevăt și ce este fals?

Hai să schimbăm perspectiva iar!

Dacă afară este întuneric atunci,
dacă porcii zboară atunci este întuneric afară.

$$\begin{aligned} A &= \text{afară este întuneric} \\ B &= \text{porcii zboară} \end{aligned} \qquad A \supset (B \supset A)$$

Este adevărată această afirmație? Da!

A	B	$B \supset A$	$A \supset (B \supset A)$
false	false	true	true
false	true	false	true
true	false	true	true
true	true	true	true

Semantica unei logici

Dăm valori variabilelor în mulțimea $\{0, 1\}$,
definim o evaluare $e : V \to \{0, 1\}$.

Putem să o extindem o evaluare la formule:

$\wedge : \{0, 1\} \times \{0, 1\} \to \{0, 1\}$

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

$\supset : \{0, 1\} \times \{0, 1\} \to \{0, 1\}$

A	B	$A \supset B$
0	0	1
0	1	1
1	0	0
1	1	1

Dacă pentru toate evaluările posibile, o formulă are valoarea 1, atunci spunem că este o tautologie.

Sintaxa unei logici

Dăm metode pentru a manipula simbolurile din logică (i.e., $\supset$, $\wedge$) pentru a stabili când o formulă este demonstrabilă/teoremă .

Corectitudine	=	sintaxa implică semantica
Completitudine	=	sintaxa și semantica coincid

Un sistem de deducție naturală

Reguli pentru a manevra fiecare conector logic (introducerea și eliminarea conectorilor).

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\ (\wedge_I) \qquad \frac{\vdash A \wedge B}{\vdash A}\ (\wedge_{E_1}) \qquad \frac{\vdash A \wedge B}{\vdash B}\ (\wedge_{E_2})$$

$$\frac{\{A\} \vdash B}{\vdash A \supset B}\ (\supset_I) \qquad \frac{\vdash A \supset B \qquad \vdash A}{\vdash B}\ (\supset_E)$$

Arată cunoscut?

Corespondența Curry-Howard

λ-calcul cu tipuri	Deducție naturală
$\dfrac{\vdash M : A \qquad \vdash N : B}{\vdash \langle M, N\rangle : A \times B}\ (\times_I)$	$\dfrac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\ (\wedge_I)$
$\dfrac{\vdash M : A \times B}{\vdash fst\ M : A}\ (\times_{E_1})$	$\dfrac{\vdash A \wedge B}{\vdash A}\ (\wedge_{E_1})$
$\dfrac{\vdash M : A \times B}{\vdash snd\ M : B}\ (\times_{E_2})$	$\dfrac{\vdash A \wedge B}{\vdash B}\ (\wedge_{E_2})$
$\dfrac{\{x : A\} \vdash M : B}{\vdash \lambda x.M : A \to B}\ (\to_I)$	$\dfrac{\{A\} \vdash B}{\vdash A \supset B}\ (\supset_I)$
$\dfrac{\vdash M : A \to B \qquad \vdash N : A}{\vdash M\,N : B}\ (\to_E)$	$\dfrac{\vdash A \supset B \qquad \vdash A}{\vdash B}\ (\supset_E)$

Propositions are types! ♡

Să analizăm mai atent

λ-calcul cu tipuri	Deducție naturală
$\Gamma \vdash M : A$	$\Gamma \vdash A$

Faptul că există un termen de tip A *(inhabitation of type A)*
înseamnă că A este teoremă/are o demonstrație în logică! ♡

Să analizăm mai atent

λ-calcul cu tipuri

$$\frac{\{x : A\} \vdash x : A}{\vdash \lambda x.\, x : A \to A}\ (\to_I)$$

Deducție naturală

$$\frac{\{A\} \vdash A}{\vdash A \supset A}\ (\supset_I)$$

Să analizăm mai atent

λ-calcul cu tipuri

$$\frac{\{x : A\} \vdash x : A}{\vdash \lambda x.\, x : A \to A}\ (\to_I)$$

$$\cfrac{\cfrac{\overline{\{x : A, y : B\} \vdash x : A}}{\{x : A\} \vdash \lambda y.\, x : B \to A}\ (\to_I)}{\vdash \lambda x.\, (\lambda y.\, x) : A \to (B \to A)}\ (\to_I)$$

Deducție naturală

$$\frac{\{A\} \vdash A}{\vdash A \supset A}\ (\supset_I)$$

$$\cfrac{\cfrac{\overline{\{A, B\} \vdash A}}{\{A\} \vdash B \to A}\ (\supset_I)}{\vdash A \to (B \to A)}\ (\supset_I)$$

Să analizăm mai atent

λ-calcul cu tipuri

$$\frac{\{x : A\} \vdash x : A}{\vdash \lambda x.\, x : A \to A}\ (\to_I)$$

$$\cfrac{\cfrac{\overline{\{x : A, y : B\} \vdash x : A}}{\{x : A\} \vdash \lambda y.\, x : B \to A}\ (\to_I)}{\vdash \lambda x.\, (\lambda y.\, x) : A \to (B \to A)}\ (\to_I)$$

Deducție naturală

$$\frac{\{A\} \vdash A}{\vdash A \supset A}\ (\supset_I)$$

$$\cfrac{\cfrac{\overline{\{A, B\} \vdash A}}{\{A\} \vdash B \to A}\ (\supset_I)}{\vdash A \to (B \to A)}\ (\supset_I)$$

Proofs are Terms! ♡

Demonstrațiile sunt termeni!

Corespondența Curry-Howard

Teoria Tipurilor	Logică
tipuri	formule
termeni	demonstrații
inhabitation a tipului A	demonstrație a lui A

Corespondența Curry-Howard

Teoria Tipurilor	Logică
tipuri	formule
termeni	demonstrații
inhabitation a tipului A	demonstrație a lui A
tip produs	conjuncție
tip funcție	implicație

Corespondența Curry-Howard

Teoria Tipurilor	Logică
tipuri	formule
termeni	demonstrații
inhabitation a tipului A	demonstrație a lui A
tip produs	conjuncție
tip funcție	implicație
tip sumă	disjuncție
tipul void	false
tipul unit	true

Logica intuiționistă

- Logică constructivistă
- Bazată pe noțiunea de demonstrație
- Utilă deoarece demonstrațiile sunt executabile și produc exemple
  Permite "extragererea" de programe demonstrate a fi corecte.
- Baza pentru *proof assistants* (e.g., Coq, Lean, Agda, Idris)
- Următoarele formule echivalente nu sunt demonstrabile în logica intuiționistă!
  - dubla negație: $\neg\neg\varphi \supset \varphi$
  - excluded middle: $\varphi \vee \neg\varphi$
  - legea lui Pierce: $((\varphi \supset \psi) \supset \varphi) \supset \varphi$
- Nu există semantică cu tabele de adevăr pentru logica intuiționistă! Semantici alternative (e.g., semantica de tip Kripke)

Corespondența Curry-Howard

Inițial, corespondența Curry-Howard a fost între

Calculul Church $\lambda\rightarrow$	Sistemul de deducție naturală al lui Gentzen pentru logica intuiționistă

De ce?

- Este pur si simplu fascinant
- Nu gândiți logica și informatica ca domenii diferite.
- Gândind din perspective diferite ne poate ajuta să știm ce este posibil/imposibil.
- Teoria tipurilor nu ar trebui să fie o adunătură *ad hoc* de reguli!