Curs 14

2021-2022

Fundamentele limbajelor de programare

Cuprins

Examen

Informatii generale despre examen

- 24 iunie
- 2 ore, fizic, în laboratoare/amfiteatre
- cu materialele ajutătoare de la curs/seminar/laborator
- 1 punct din oficiu
- condiţia minimă pentru a promova: nota examen > 4.99

Informatii generale despre examen

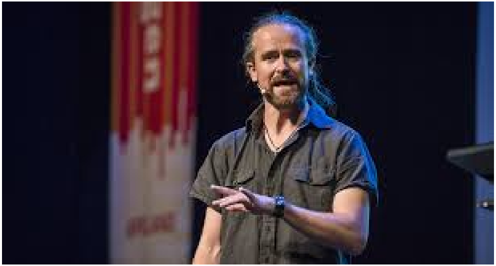
- Trebuie să rezolvaţi atât probleme pe calculator, cât şi pe foaie
- Dacă doriţi să susţineţi examenul pe laptop-ul personal, vă rugăm să completaţi formularul de mai jos (primul venit, primul servit, în funcţie de numărul de locuri disponibile):
  - seria 23:
    `https://tinyurl.com/4t4y74aa`
  - seria 24:
    `https://tinyurl.com/5tu5x2h7`
  - seria 25:
    `https://tinyurl.com/3ymm9hsz`
- Termen limită pentru completarea formularelor: **2.06.2022**

Structură examen

- Partea teoretică (4 puncte) - 3 probleme din lista de mai jos:
  - unificare
  - deducţie naturală
  - puncte fixe
  - rezoluţie
  - arbori de execuţie şi arbori SLD
  - paşi în semantica operaţională
  - substituţii şi β-reduceri în lambda calcul
- Partea practică (5 puncte)
  1. o problemă tipică de Prolog (2 puncte)
  2. o problemă de limbaj de programare (3 puncte)
     - se dă sintaxa unui limbaj de programare
     - să se verifice dacă un şir de caractere este un program corect în limbaj
     - să se implementeze un interpretor pentru limbaj

Corespondenţa Curry-Howard-Lambek

Schimbaţi perspectiva



Roger Antonsen
Universitatea din Oslo

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

... inţelegerea este legată de abilitatea de a-ţi schimba perspectiva.

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

Corespondenţa Curry-Howard-Lambek

Ne vom uita la nişte concepte din trei perspective diferite:

- **Teoria Tipurilor**
- **Logică**
- **Teoria Categoriilor**

Un program simplu în 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!

Un program simplu în Haskell

Hai să schimbăm perspectiva!

```
data Point = Point Int Int

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

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

Un program simplu în Haskell

Hai să schimbăm perspectiva!

```
data Point = Point Int Int

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

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

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

$$\frac{p : Point}{getX\ p : Int}$$

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

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

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

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

$$\frac{p : Point}{getX\ p : Int}$$

$$\frac{p : Point}{getY\ p : Int}$$

Un program simplu în Haskell

Hai să schimbăm perspectiva!

```
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
```

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

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

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

Generalizare

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

Generalizare

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

$$\frac{a : A \quad b : B}{\langle a, b \rangle : A \times B}\ (\times_I)$$

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

$$\frac{p : A \times B}{fst\ p : A}\ (\times_{E_1})$$

Generalizare

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

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

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

Alt exemplu simplu

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

Alt exemplu simplu

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

$$\dfrac{\begin{array}{c}[x : Int]\\ \vdots \\ x * 3 : Int\end{array}}{\lambda x.x * 3 : Int \to Int}$$

Alt exemplu simplu

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

$$\dfrac{\begin{array}{c}[x : Int] \\ \vdots \\ x * 3 : Int\end{array}}{\lambda x.x * 3 : Int \to Int}$$

```
> f 5
15
```

Alt exemplu simplu

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

$$\dfrac{\begin{array}{c}[x : Int] \\ \vdots \\ x * 3 : Int\end{array}}{\lambda x.x * 3 : Int \to Int}$$

```
> f 5
15
```

$$\dfrac{f : Int \to Int \quad 5 : Int}{f\,5 : Int}$$

Alt exemplu simplu

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

$$\dfrac{\begin{array}{c}[x : Int]\\ \vdots \\ x * 3 : Int\end{array}}{\lambda x.x * 3 : Int \to Int}\ (fun_I)$$

```
> f 5
15
```

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

Generalizare

$$\dfrac{\begin{matrix}[x : Int] \\ \vdots \\ x * 3 : Int\end{matrix}}{\lambda x.x * 3 : Int \to Int}\ (fun_I)$$

$$\dfrac{\begin{matrix}[x : A] \\ \vdots \\ b : B\end{matrix}}{\lambda x.b : A \to B}\ (\to_I)$$

Generalizare

$$\dfrac{\begin{array}{c}[x : Int] \\ \vdots \\ x * 3 : Int\end{array}}{\lambda x.x * 3 : Int \to Int}\ (fun_I) \qquad \dfrac{\begin{array}{c}[x : A] \\ \vdots \\ b : B\end{array}}{\lambda x.b : A \to B}\ (\to_I)$$

$$\dfrac{f : Int \to Int \quad 5 : Int}{f\ 5 : Int}\ (fun_E) \qquad \dfrac{f : A \to B \quad x : A}{f\ x : B}\ (\to_E)$$

Generalizare

$$\frac{\begin{matrix}[x : Int] \\ \vdots \\ x * 3 : Int\end{matrix}}{\lambda x.x * 3 : Int \rightarrow Int}\ (fun_I)$$

$$\frac{\begin{matrix}[x : A] \\ \vdots \\ b : B\end{matrix}}{\lambda x.b : A \rightarrow B}\ (\rightarrow_I)$$

(abstracţie)

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

$$\frac{f : A \rightarrow B \quad x : A}{f\ x : B}\ (\rightarrow_E)$$

(aplicare)

Un λ-calcul cu tipuri

$$\frac{a : A \quad b : B}{\langle a, b\rangle : A \times B}\ (\times_I)$$

$$\frac{p : A \times B}{\mathit{fst}\ p : A}\ (\times_{E_1})$$

$$\frac{p : A \times B}{\mathit{snd}\ p : B}\ (\times_{E_2})$$

$$\frac{\begin{array}{c}[x : A]\\ \vdots\\ b : B\end{array}}{\lambda x.b : A \to B}\ (\to_I)$$

$$\frac{f : A \to B \quad x : A}{f\ x : B}\ (\to_E)$$

Logica. Ce este adevăt şi ce este fals?

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

Logica. Ce este adevăt şi ce este fals?

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 şi ce este fals?

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

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

Este adevărată această afirmaţie?

Logica. Ce este adevăt şi ce este fals?

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

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

Este adevărată această afirmaţie?

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

Logica. Ce este adevăt şi ce este fals?

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

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

Este adevărată această afirmaţie?

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

Da!

Semantica

Dăm valori atomilor în mulţimea $\{0, 1\}$, definim o evaluare $e : Atoms \rightarrow \{0, 1\}$.

Semantica

Dăm valori atomilor în mulţimea $\{0, 1\}$, definim o evaluare $e : Atoms \to \{0, 1\}$.

Având o evaluare, putem să o extindem la formule folosind tabelele de adevăr:

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

A	B	$A \& B$
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

Semantica

Dăm valori atomilor în mulţimea $\{0,1\}$, definim o evaluare $e : Atoms \to \{0,1\}$.

Având o evaluare, putem să o extindem la formule folosind tabelele de adevăr:

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

A	B	$A \& B$
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 de adevăr 1, atunci spunem că este mereu adevărată (este o tautologie).

Logică

Sintaxa unei logici

- Noţiunile de teoremă şi demonstrabilitate
- Oferă metode de a manipula simboluri din logică (i.e., atomi, $\supset$, &) pentru a stabili când o formulă este demonstrabilă (aka este teoremă).

Logică

Sintaxa unei logici

- Noţiunile de teoremă şi demonstrabilitate
- Oferă metode de a manipula simboluri din logică (i.e., atomi, $\supset$, &) pentru a stabili când o formulă este demonstrabilă (aka este teoremă).

Completitudine	=	sintaxa şi semantica coincid
Corectitudine	=	sintaxa implică semantica

Un sistem de deducţie naturală

- Reguli pentru a manevra fiecare conector logic
- Reguli pentru introducerea si eliminarea conectorilor
- Reguli de forma

$$\frac{\textit{Ipoteze}}{\textit{Concluzie}}$$

Un sistem de deducţie naturală

- Reguli pentru a manevra fiecare conector logic
- Reguli pentru introducerea si eliminarea conectorilor
- Reguli de forma

$$\frac{\textit{Ipoteze}}{\textit{Concluzie}}$$

$$\frac{A \quad B}{A \& B}\ (\&_I)$$

Un sistem de deducţie naturală

- Reguli pentru a manevra fiecare conector logic
- Reguli pentru introducerea si eliminarea conectorilor
- Reguli de forma

$$\frac{Ipoteze}{Concluzie}$$

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad \frac{A\&B}{A}\ (\&_{E_1}) \qquad \frac{A\&B}{B}\ (\&_{E_2})$$

Un sistem de deducţie naturală

- Reguli pentru a manevra fiecare conector logic
- Reguli pentru introducerea si eliminarea conectorilor
- Reguli de forma

$$\frac{\textit{Ipoteze}}{\textit{Concluzie}}$$

$$\frac{A \quad B}{A \& B}\ (\&_I) \qquad \frac{A \& B}{A}\ (\&_{E_1}) \qquad \frac{A \& B}{B}\ (\&_{E_2})$$

$$\frac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \supset B}\ (\supset_I)$$

Un sistem de deducţie naturală

- Reguli pentru a manevra fiecare conector logic
- Reguli pentru introducerea si eliminarea conectorilor
- Reguli de forma

$$\frac{\mathit{Ipoteze}}{\mathit{Concluzie}}$$

$$\frac{A \quad B}{A \& B}\ (\&_I) \qquad \frac{A \& B}{A}\ (\&_{E_1}) \qquad \frac{A \& B}{B}\ (\&_{E_2})$$

$$\frac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \supset B}\ (\supset_I) \qquad \frac{A \supset B \quad A}{B}\ (\supset_E)$$

Un sistem de deducţie naturală

- Reguli pentru a manevra fiecare conector logic
- Reguli pentru introducerea si eliminarea conectorilor
- Reguli de forma

$$\frac{\textit{Ipoteze}}{\textit{Concluzie}}$$

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad \frac{A\&B}{A}\ (\&_{E_1}) \qquad \frac{A\&B}{B}\ (\&_{E_2})$$

$$\frac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \supset B}\ (\supset_I) \qquad \frac{A \supset B \quad A}{B}\ (\supset_E)$$

Arată cunoscut?

Ce am văzut până acum

Un λ-calcul cu tipuri	Un sistem de deducţie naturală
$\dfrac{a : A \quad b : B}{\langle a, b\rangle : A \times B}\ (\times_I)$	$\dfrac{A \quad B}{A\&B}\ (\&_I)$
$\dfrac{p : A \times B}{fst\ p : A}\ (\times_{E_1})$	$\dfrac{A\&B}{A}\ (\&_{E_1})$
$\dfrac{p : A \times B}{snd\ p : B}\ (\times_{E_2})$	$\dfrac{A\&B}{B}\ (\&_{E_2})$
$\dfrac{\begin{matrix}[x : A] \\ \vdots \\ b : B\end{matrix}}{\lambda x.n : A \to B}\ (\to_I)$	$\dfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \supset B}\ (\supset_I)$
$\dfrac{f : A \to B \quad x : A}{f\ x : B}\ (\to_E)$	$\dfrac{A \supset B \quad A}{B}\ (\supset_E)$

Ce am văzut până acum

Un λ-calcul cu tipuri | Un sistem de deducţie naturală

$$\frac{a : A \quad b : B}{\langle a, b\rangle : A \times B}\ (\times_I) \qquad \frac{A \quad B}{A \& B}\ (\&_I)$$

$$\frac{p : A \times B}{fst\ p : A}\ (\times_{E_1}) \qquad \frac{A \& B}{A}\ (\&_{E_1})$$

$$\frac{p : A \times B}{snd\ p : B}\ (\times_{E_2}) \qquad \frac{A \& B}{B}\ (\&_{E_2})$$

$$\begin{array}{c} [x : A] \\ \vdots \\ b : B \\ \hline \lambda x.n : A \to B \end{array}\ (\to_I) \qquad \begin{array}{c} [A] \\ \vdots \\ B \\ \hline A \supset B \end{array}\ (\supset_I)$$

$$\frac{f : A \to B \quad x : A}{f\ x : B}\ (\to_E) \qquad \frac{A \supset B \quad A}{B}\ (\supset_E)$$

Propositions are types! ♡

Să analizăm mai atent

Un λ-calcul cu tipuri	Un sistem de deducţie naturală
$\dfrac{a : A \quad b : B}{\langle a, b\rangle : A \times B}\ (\times_I)$	$\dfrac{A \quad B}{A \& B}\ (\&_I)$

Să analizăm mai atent

Un λ-calcul cu tipuri	Un sistem de deducţie naturală
$\dfrac{a : A \quad b : B}{\langle a, b \rangle : A \times B}\ (\times_I)$	$\dfrac{A \quad B}{A \& B}\ (\&_I)$

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

Să analizăm mai atent

Un λ-calcul cu tipuri

Un sistem de deducţie naturală

$: A \to A$

Să analizăm mai atent

Un λ-calcul cu tipuri | Un sistem de deducţie naturală

$\lambda x.x : A \to A$ (id)

Să analizăm mai atent

Un λ-calcul cu tipuri

$\lambda x.x : A \to A$ (id)

Un sistem de deducţie naturală

$$\frac{}{A \supset A}$$

Să analizăm mai atent

Un λ-calcul cu tipuri

$\lambda x.x : A \to A$ (id)

Un sistem de deducţie naturală

$$\frac{[A]}{A \supset A}$$

Să analizăm mai atent

Un λ-calcul cu tipuri

$\lambda x.x : A \to A$ (id)

Un sistem de deducţie naturală

$$\dfrac{\begin{matrix}[A] \\ A\end{matrix}}{A \supset A}$$

Să analizăm mai atent

Un λ-calcul cu tipuri

$\lambda x.x : A \to A \quad \text{(id)}$

$: A \to (B \to A)$

Un sistem de deducţie naturală

$$\frac{\begin{array}{c}[A]\\ A\end{array}}{A \supset A}$$

Să analizăm mai atent

Un λ-calcul cu tipuri	Un sistem de deducţie naturală
$\lambda x.x : A \rightarrow A$ (id)	$$\dfrac{\begin{array}{c}[A]\\ A\end{array}}{A \supset A}$$
$\lambda x.(\lambda y.x) : A \rightarrow (B \rightarrow A)$ (const)	

Să analizăm mai atent

Un λ-calcul cu tipuri	Un sistem de deducţie naturală
$\lambda x.x : A \to A$ (id)	$\dfrac{\begin{array}{c}[A]\\ A\end{array}}{A \supset A}$
$\lambda x.(\lambda y.x) : A \to (B \to A)$ (const)	$\dfrac{}{A \supset (B \supset A)}$

Să analizăm mai atent

Un λ-calcul cu tipuri

Un sistem de deducţie naturală

$\lambda x.x : A \to A$ (id)

$$\frac{\begin{matrix}[A] \\ A\end{matrix}}{A \supset A}$$

$\lambda x.(\lambda y.x) : A \to (B \to A)$ (const)

$$\frac{\begin{matrix}[A] \\ \\ B \supset A\end{matrix}}{A \supset (B \supset A)}$$

Să analizăm mai atent

Un λ-calcul cu tipuri	Un sistem de deducţie naturală
$\lambda x.x : A \to A$ (id)	$$\dfrac{\begin{matrix}[A]\\ A\end{matrix}}{A \supset A}$$
$\lambda x.(\lambda y.x) : A \to (B \to A)$ (const)	$$\dfrac{\dfrac{\begin{matrix}[A]\\ [B]\end{matrix}}{B \supset A}}{A \supset (B \supset A)}$$

Să analizăm mai atent

Un λ-calcul cu tipuri

Un sistem de deducţie naturală

$\lambda x.x : A \to A$ (id)

$$\dfrac{\begin{matrix}[A] \\ A\end{matrix}}{A \supset A}$$

$\lambda x.(\lambda y.x) : A \to (B \to A)$ (const)

$$\dfrac{\dfrac{\begin{matrix}[A] \\ [B] \\ A\end{matrix}}{B \supset A}}{A \supset (B \supset A)}$$

Să analizăm mai atent

Un λ-calcul cu tipuri	Un sistem de deducţie naturală
$\lambda x.x : A \to A$ (id)	$$\dfrac{\begin{array}{c}[A]\\ A\end{array}}{A \supset A}$$
$\lambda x.(\lambda y.x) : A \to (B \to A)$ (const)	$$\dfrac{\dfrac{\begin{array}{c}[A]\\ [B]\\ A\end{array}}{B \supset A}}{A \supset (B \supset A)}$$

Proofs are Terms! ♡

Demonstraţiile sunt termeni!

Corespondenţa Curry-Howard

Teoria Tipurilor	Logică
tipuri	formule
termeni	demonstraţii
inhabitation a tipului A	demonstraţie a lui A

Corespondenţa Curry-Howard

Teoria Tipurilor	Logică
tipuri	formule
termeni	demonstraţii
inhabitation a tipului A	demonstraţie a lui A
tip produs	conjuncţie
tip funcţie	implicaţie

Corespondenţa Curry-Howard

Teoria Tipurilor	Logică
tipuri	formule
termeni	demonstraţii
inhabitation a tipului A	demonstraţie a lui A
tip produs	conjuncţie
tip funcţie	implicaţie
tip sumă	disjuncţie
tipul void	false
tipul unit	true

Deducţie naturală pentru logica clasică

$$\dfrac{\begin{array}{c}[A]\\ \vdots\\ B\end{array}}{A \supset B}\ (\supset_I)$$

$$\dfrac{A \supset B \quad A}{B}\ (\supset_E)$$

$$\dfrac{A \quad B}{A \& B}\ (\&_I)$$

$$\dfrac{A \& B}{A}\ (\&_{E_1})$$

$$\dfrac{A \& B}{B}\ (\&_{E_2})$$

Deducţie naturală pentru logica clasică

$$\dfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \supset B}\ (\supset_I) \qquad \dfrac{A \supset B \quad A}{B}\ (\supset_E)$$

$$\dfrac{A \quad B}{A \& B}\ (\&_I) \qquad \dfrac{A \& B}{A}\ (\&_{E_1}) \qquad \dfrac{A \& B}{B}\ (\&_{E_2})$$

$$\dfrac{A}{A \vee B}\ (\vee_{I_1}) \qquad \dfrac{B}{A \vee B}\ (\vee_{I_2}) \qquad \dfrac{A \vee B \quad A \supset C \quad B \supset C}{C}\ (\vee_E)$$

Deducţie naturală pentru logica clasică

$$\dfrac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \supset B}\ (\supset_I) \qquad \dfrac{A \supset B \quad A}{B}\ (\supset_E)$$

$$\dfrac{A \quad B}{A \& B}\ (\&_I) \qquad \dfrac{A \& B}{A}\ (\&_{E_1}) \qquad \dfrac{A \& B}{B}\ (\&_{E_2})$$

$$\dfrac{A}{A \vee B}\ (\vee_{I_1}) \qquad \dfrac{B}{A \vee B}\ (\vee_{I_2}) \qquad \dfrac{A \vee B \quad A \supset C \quad B \supset C}{C}\ (\vee_E)$$

$$\dfrac{\perp}{A}\ \text{(ex falso quodlibet)}$$

Deducţie naturală pentru logica clasică

$$\dfrac{\begin{array}{c}[A]\\ \vdots\\ B\end{array}}{A \supset B}\ (\supset_I) \qquad \dfrac{A \supset B \quad A}{B}\ (\supset_E)$$

$$\dfrac{A \quad B}{A\&B}\ (\&_I) \qquad \dfrac{A\&B}{A}\ (\&_{E_1}) \qquad \dfrac{A\&B}{B}\ (\&_{E_2})$$

$$\dfrac{A}{A \vee B}\ (\vee_{I_1}) \qquad \dfrac{B}{A \vee B}\ (\vee_{I_2}) \qquad \dfrac{A \vee B \quad A \supset C \quad B \supset C}{C}\ (\vee_E)$$

$$\dfrac{\perp}{A}\ \text{(ex falso quodlibet)} \qquad \boxed{\dfrac{\neg\neg A}{A}\ \text{(reductio ad absurdum)}}$$

($\neg A$ este o abreviere pentru $A \supset \perp$)

Deducţie naturală pentru logica intuiţionistă

$$\frac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \supset B}\ (\supset_I) \qquad \frac{A \supset B \quad A}{B}\ (\supset_E)$$

$$\frac{A \quad B}{A \& B}\ (\&_I) \qquad \frac{A \& B}{A}\ (\&_{E_1}) \qquad \frac{A \& B}{B}\ (\&_{E_2})$$

$$\frac{A}{A \vee B}\ (\vee_{I_1}) \qquad \frac{B}{A \vee B}\ (\vee_{I_2}) \qquad \frac{A \vee B \quad A \supset C \quad B \supset C}{C}\ (\vee_E)$$

$$\frac{\perp}{A}\ (\text{ex falso quodlibet})$$

Logica intuiţionistă

- Logică constructivistă
- Bazată pe noţiunea de demonstraţie
- Utilă deoarece demonstraţiile sunt executabile şi produc exemple
- Baza pentru *proof assistants* (e.g., Coq, Lean, Isabele, Agda, Idris)

Logica intuiţionistă

- Logică constructivistă
- Bazată pe noţiunea de demonstraţie
- Utilă deoarece demonstraţiile sunt executabile şi produc exemple
- Baza pentru *proof assistants* (e.g., Coq, Lean, Isabele, Agda, Idris)
- Următoarele formule echivalente nu sunt demonstrabile în logica intuiţionistă:
  - dubla negaţie: $\neg\neg A \supset A$
  - excluded middle: $A \vee \neg A$
  - legea lui Pierce: $((A \supset B) \supset A) \supset A$

Logica intuiţionistă

- Logică constructivistă
- Bazată pe noţiunea de demonstraţie
- Utilă deoarece demonstraţiile sunt executabile şi produc exemple
- Baza pentru *proof assistants* (e.g., Coq, Lean, Isabele, Agda, Idris)
- Următoarele formule echivalente nu sunt demonstrabile în logica intuiţionistă:
  - dubla negaţie: $\neg\neg A \supset A$
  - excluded middle: $A \vee \neg A$
  - legea lui Pierce: $((A \supset B) \supset A) \supset A$
- Nu există semantică cu tabele de adevăr pentru logica intuiţionistă!
  Are semantici alternative (e.g., semantica de tip Kripke)

Logica intuiţionistă

- Logică constructivistă
- Bazată pe noţiunea de demonstraţie
- Utilă deoarece demonstraţiile sunt executabile şi produc exemple
- Baza pentru *proof assistants* (e.g., Coq, Lean, Isabele, Agda, Idris)
- Următoarele formule echivalente nu sunt demonstrabile în logica intuiţionistă:
  - dubla negaţie: $\neg\neg A \supset A$
  - excluded middle: $A \vee \neg A$
  - legea lui Pierce: $((A \supset B) \supset A) \supset A$
- Nu există semantică cu tabele de adevăr pentru logica intuiţionistă!
  Are semantici alternative (e.g., semantica de tip Kripke)

Iniţial, Corespondenţa Curry-Howard a fost între

Church's simply typed λ-calculus şi Gentzen's natural deduction for intuitionistic logic.

Corespondenţa Curry-Howard în general

Teoria Tipurilor	Logică
tipuri	formule
termeni	demonstraţii
inhabitation a tipului A	demonstraţie a lui A
tip produs	conjuncţie
tip funcţie	implicaţie
tip sumă	disjuncţie
tipul void	false
tipul unit	true
dependent types	cuantificatori
`call/cc` operator	Peirce's law
monade	o logică modală
...	...

De ce?

- Este pur si simplu fascinant

De ce?

- Este pur si simplu fascinant
- Nu gândiţi logica şi informatica ca domenii diferite.

De ce?

- Este pur si simplu fascinant
- Nu gândiţi logica şi informatica ca domenii diferite.
- Gândind din perspective diferite ne poate ajuta să ştim ce este posibil/imposibil.

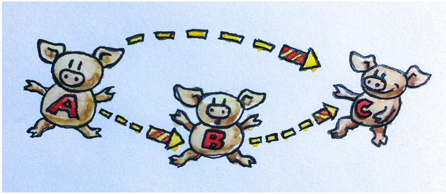
De ce?

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

Să schimbăm iar perspectiva!

Teoria categoriilor

- A category is an embarrassingly simple concept.
  Bartosz Milewski, Category Theory for Programmers
- Categorie = obiecte + săgeţi
- Ingredient cheie: compunerea de săgeţi



credits: Bartosz Milewski

O categorie

O categorie **C** constă în

- Obiecte:
- Săgeţi:
- Compunere:

O categorie

O categorie **C** constă în

- Obiecte: notate $A, B, C, \ldots$
- Săgeţi:
- Compunere:

O categorie

O categorie **C** constă în

- Obiecte: notate $A, B, C, \ldots$
- Săgeţi: pentru orice obiecte A şi B, există o mulţime de săgeţi $\mathbf{C}(A, B)$
  - notăm $f \in \mathbf{C}(A, B)$ cu $f : A \to B$ sau $A \xrightarrow{f} B$
- Compunere:

O categorie

O categorie **C** constă în

- Obiecte: notate $A, B, C, \ldots$
- Săgeţi: pentru orice obiecte A şi B, există o mulţime de săgeţi $\mathbf{C}(A, B)$
  - notăm $f \in \mathbf{C}(A, B)$ cu $f : A \to B$ sau $A \xrightarrow{f} B$
- Compunere: pentru orice săgeţi $f : A \to B$ şi $g : B \to C$ există o săgeată $g \circ f : A \to C$

O categorie

O categorie $\mathbf{C}$ constă în

- Obiecte: notate $A, B, C, \ldots$
- Săgeţi: pentru orice obiecte A şi B, există o mulţime de săgeţi $\mathbf{C}(A, B)$
  - notăm $f \in \mathbf{C}(A, B)$ cu $f : A \to B$ sau $A \xrightarrow{f} B$
- Compunere: pentru orice săgeţi $f : A \to B$ şi $g : B \to C$ există o săgeată $g \circ f : A \to C$

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, dashed, "g \circ f"'] & B \arrow[d, "g"] \\ & C \end{tikzcd}$$

O categorie

O categorie **C** constă în

- Obiecte: notate $A, B, C, \ldots$
- Săgeţi: pentru orice obiecte A şi B, există o mulţime de săgeţi $\mathbf{C}(A, B)$
  - notăm $f \in \mathbf{C}(A, B)$ cu $f : A \to B$ sau $A \xrightarrow{f} B$
- Compunere: pentru orice săgeţi $f : A \to B$ şi $g : B \to C$ există o săgeată $g \circ f : A \to C$

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, dashed, "g \circ f"'] & B \arrow[d, "g"] \\ & C \end{tikzcd}$$

- Identitate: pentru orice obiect A există o săgeată $id_A : A \to A$

O categorie

O categorie $\mathbf{C}$ constă în

- Obiecte: notate $A, B, C, \ldots$
- Săgeţi: pentru orice obiecte A şi B, există o mulţime de săgeţi $\mathbf{C}(A, B)$
  - notăm $f \in \mathbf{C}(A, B)$ cu $f : A \to B$ sau $A \xrightarrow{f} B$
- Compunere: pentru orice săgeţi $f : A \to B$ şi $g : B \to C$ există o săgeată $g \circ f : A \to C$

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, dashed, "g \circ f"'] & B \arrow[d, "g"] \\ & C \end{tikzcd}$$

- Identitate: pentru orice obiect A există o săgeată $id_A : A \to A$
- Axiome: pentru orice săgeţi $f : A \to B$, $g : B \to C$ şi $h : C \to D$

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad f \circ id_A = f = id_B \circ f$$

Exemplu - categoria de mulţimi

Categoria **Set** are

- Obiecte: mulţimi
- Săgeţi: funcţii
- Compunere: compunerea de funcţii
- Identitate: pentru orice mulţime A, funcţia identitate $id_A : A \to A$, $id_A(a) = a$
- Axiome: ✓

Exemplu - categoria de monoizi

Categoria **Mon** are

- Obiecte: monoizi
- Săgeţi: morfisme de monoizi
  (aka funcţii care nu "strică" operaţia de monoid)
- Compunerea: compunerea de morfisme de monoizi
- Identitatea: pentru orice obiect **M**, $id_{\mathbf{M}} : M \to M$, $id_M(m) = m$
- Axiome: ✓

Exemplu - un monoid ca o categorie

Orice monoid $\mathbf{M} = \langle M, +, e \rangle$ este o categorie cu

- Obiecte: un singur obiect $\square$
- Săgeţi: elementele mulţimii M (i.e, $\mathbf{M}(\square, \square) = M$)
- Compunerea: operaţia de monoid $+$
- Identitatea: identitatea monoidului e
- Axiome:

$$\begin{array}{cc} h \circ (g \circ f) = (h \circ g) \circ f & f \circ id_A = f = id_B \circ f \\ a + (b + c) = (a + b) + c & a + e = a = e + a \end{array}$$

Obiect terminal şi obiect iniţial

Într-o categorie **C**

- un obiect T se numeşte terminal dacă pentru orice obiect A există o unică săgeată

$$\tau_A : A \to T$$

Obiect terminal şi obiect iniţial

Într-o categorie **C**

- un obiect T se numeşte terminal dacă pentru orice obiect A există o unică săgeată
$$\tau_A : A \to T$$

- un obiect I se numeşte iniţial dacă pentru orice obiect A există o unică săgeată
$$\iota_A : I \to A$$

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A \& B}\ (\&_I)$$

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A\&B}\ (\&_I)$$

Deducţie din ipotezele Γ

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A\&B}\ (\&_I)$$

Deducţie fără ipoteze $\Gamma = \emptyset$ — Deducţie din ipotezele Γ

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A \& B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B}\ (\&_I)$$

Deducţie fără ipoteze
$\Gamma = \emptyset$ (Deducţie din adevăr)

Deducţie din ipotezele Γ

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A \& B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B}\ (\&_I)$$

Deducţie fără ipoteze
$\Gamma = \emptyset$ (Deducţie din adevăr)

Deducţie din ipotezele Γ

Fie **C** o categorie cu obiect terminal T.

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A\&B}\ (\&_I)$$

Deducţie fără ipoteze
$\Gamma = \emptyset$ (Deducţie din adevăr)

Deducţie din ipotezele Γ

Fie **C** o categorie cu obiect terminal T. Avem următoarele interpretări:

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A\&B}\ (\&_I)$$

Deducţie fără ipoteze
$\Gamma = \emptyset$ (Deducţie din adevăr)

Deducţie din ipotezele Γ

Fie **C** o categorie cu obiect terminal T. Avem următoarele interpretări:

- Formulele sunt obiectele din **C**

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A\&B}\ (\&_I)$$

Deducţie fără ipoteze
$\Gamma = \emptyset$ (Deducţie din adevăr)

Deducţie din ipotezele Γ

Fie **C** o categorie cu obiect terminal T. Avem următoarele interpretări:

- Formulele sunt obiectele din **C**
- Adevărul este obiectul terminal T

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A\&B}\ (\&_I)$$

Deducţie fără ipoteze
$\Gamma = \emptyset$ (Deducţie din adevăr)

Deducţie din ipotezele Γ

Fie **C** o categorie cu obiect terminal T. Avem următoarele interpretări:

- Formulele sunt obiectele din **C**
- Adevărul este obiectul terminal T
- O demonstraţie a lui A este o săgeată $f : T \to A$

De ce obiect terminal?

Putem generaliza regulile de mai devreme. De exemplu:

$$\frac{A \quad B}{A\&B}\ (\&_I) \qquad\qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A\&B}\ (\&_I)$$

Deducţie fără ipoteze
$\Gamma = \emptyset$ (Deducţie din adevăr)

Deducţie din ipotezele Γ

Fie **C** o categorie cu obiect terminal T. Avem următoarele interpretări:

- Formulele sunt obiectele din **C**
- Adevărul este obiectul terminal T
- O demonstraţie a lui A este o săgeată $f : T \to A$
- O demonstraţie a lui A din ipoteza B este o săgeată $f : B \to A$

Produs

Fie A şi B două obiecte în categoria **C**.

Produs

Fie A şi B două obiecte în categoria **C**. Spunem că

$$A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$$

este produsul lui A şi B

Produs

Fie A şi B două obiecte în categoria **C**. Spunem că

$$A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$$

este produsul lui A şi B dacă pentru orice

$$A \xleftarrow{f} C \xrightarrow{g} B$$

Produs

Fie A şi B două obiecte în categoria **C**. Spunem că

$$A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$$

este produsul lui A şi B dacă pentru orice

$$A \xleftarrow{f} C \xrightarrow{g} B$$

există o unică săgeată

$$\langle f, g \rangle : C \longrightarrow A \times B$$

Produs

Fie A şi B două obiecte în categoria **C**. Spunem că

$$A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$$

este produsul lui A şi B dacă pentru orice

$$A \xleftarrow{f} C \xrightarrow{g} B$$

există o unică săgeată

$$\langle f, g \rangle : C \longrightarrow A \times B$$

astfel încât

$$\pi_1 \circ \langle f, g \rangle = f \qquad \pi_2 \circ \langle f, g \rangle = g$$

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\
& C \arrow[ul, "f"] \arrow[u, dashed, "\langle f,g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Să schimbăm iar perspectiva!

Fie **C** o categorie cu obiect terminal si produse.
Fie A, B două obiecte în **C**.

Să schimbăm iar perspectiva!

Fie **C** o categorie cu obiect terminal si produse.
Fie A, B două obiecte în **C**.

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\
& T \arrow[ul, "f"] \arrow[u, dashed, "\langle f,g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Să schimbăm iar perspectiva!

Fie **C** o categorie cu obiect terminal si produse.
Fie A, B două obiecte în **C**.

$$\begin{tikzcd} A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\ & T \arrow[ul, "f"] \arrow[u, dashed, "\langle f,g \rangle"] \arrow[ur, "g"'] & \end{tikzcd}$$

$$\frac{f : T \to A \quad g : T \to B}{\langle f, g \rangle : T \to A \times B}$$

Să schimbăm iar perspectiva!

Fie **C** o categorie cu obiect terminal si produse.
Fie A, B două obiecte în **C**.

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\
& T \arrow[ul, "f"] \arrow[u, dashed, "\langle f,g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}$$

$$\frac{f : T \to A \quad g : T \to B}{\langle f, g \rangle : T \to A \times B}$$

$$\frac{\langle f, g \rangle : T \to A \times B}{\pi_1 \circ \langle f, g \rangle : T \to A}$$

Să schimbăm iar perspectiva!

Fie **C** o categorie cu obiect terminal si produse.
Fie A, B două obiecte în **C**.

$$\begin{tikzcd} A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\ & T \arrow[ul, "f"] \arrow[u, dashed, "\langle f,g \rangle"] \arrow[ur, "g"'] & \end{tikzcd}$$

$$\frac{f : T \to A \quad g : T \to B}{\langle f, g \rangle : T \to A \times B}$$

$$\frac{\langle f, g \rangle : T \to A \times B}{\pi_1 \circ \langle f, g \rangle : T \to A}$$

$$\frac{\langle f, g \rangle : T \to A \times B}{\pi_2 \circ \langle f, g \rangle : T \to B}$$

Să schimbăm iar perspectiva!

Fie **C** o categorie cu obiect terminal si produse.
Fie A, B două obiecte în **C**.

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\
& T \arrow[ul, "f"] \arrow[u, dashed, "{\langle f,g\rangle}"] \arrow[ur, "g"'] &
\end{tikzcd}$$

$$\frac{f : T \to A \quad g : T \to B}{\langle f, g\rangle : T \to A \times B}$$

$$\frac{\langle f, g\rangle : T \to A \times B}{\pi_1 \circ \langle f, g\rangle : T \to A}$$

$$\frac{\langle f, g\rangle : T \to A \times B}{\pi_2 \circ \langle f, g\rangle : T \to B}$$

Arată cunoscut?

Ce am văzut până acum

Un λ-calcul cu tipuri

$$\frac{a : A \quad b : B}{\langle a, b\rangle : A \times B}\ (\times_I)$$

$$\frac{p : A \times B}{fst\ p : A}\ (\times_{E_1})$$

$$\frac{p : A \times B}{snd\ p : B}\ (\times_{E_2})$$

Un sist. de deducţie naturală

$$\frac{A \quad B}{A\&B}\ (\&_I)$$

$$\frac{A\&B}{A}\ (\&_{E_1})$$

$$\frac{A\&B}{B}\ (\&_{E_2})$$

O categorie*

$$\frac{f : T \to A \quad g : T \to B}{\langle f, g\rangle : T \to A \times B}$$

$$\frac{\langle f, g\rangle : T \to A \times B}{\pi_1 \circ \langle f, g\rangle : T \to A}$$

$$\frac{\langle f, g\rangle : T \to A \times B}{\pi_2 \circ \langle f, g\rangle : T \to B}$$

* O categorie cu obiect terminal T şi produse

Corespondenţa Curry-Howard-Lambek

Teoria Tipurilor	Logică	Teoria categoriilor
tipuri	formule	obiecte

Corespondenţa Curry-Howard-Lambek

Teoria Tipurilor	Logică	Teoria categoriilor
tipuri	formule	obiecte
termeni	demonstraţii	săgeţi

Corespondenţa Curry-Howard-Lambek

Teoria Tipurilor	Logică	Teoria categoriilor
tipuri	formule	obiecte
termeni	demonstraţii	săgeţi
tipul unit	true	obiect terminal$T$

Corespondenţa Curry-Howard-Lambek

Teoria Tipurilor	Logică	Teoria categoriilor
tipuri	formule	obiecte
termeni	demonstraţii	săgeţi
inhabitation a tipului A	demonstraţie a lui A	săgeată $f : T \to A$
tipul unit	true	obiect terminal$T$

Corespondenţa Curry-Howard-Lambek

Teoria Tipurilor	Logică	Teoria categoriilor
tipuri	formule	obiecte
termeni	demonstraţii	săgeţi
inhabitation a tipului A	demonstraţie a lui A	săgeată $f : T \to A$
tip produs	conjuncţie	produs
tipul unit	true	obiect terminal$T$

Corespondenţa Curry-Howard-Lambek

Teoria Tipurilor	Logică	Teoria categoriilor
tipuri	formule	obiecte
termeni	demonstraţii	săgeţi
inhabitation a tipului A	demonstraţie a lui A	săgeată $f : T \to A$
tip produs	conjuncţie	produs
tip sumă	disjuncţie	coprodus
tipul unit	true	obiect terminal$T$

Corespondenţa Curry-Howard-Lambek

Teoria Tipurilor	Logică	Teoria categoriilor
tipuri	formule	obiecte
termeni	demonstraţii	săgeţi
inhabitation a tipului A	demonstraţie a lui A	săgeată $f : T \to A$
tip produs	conjuncţie	produs
tip sumă	disjuncţie	coprodus
tipul void	false	obiect iniţial
tipul unit	true	obiect terminal T

Corespondenţa Curry-Howard-Lambek

Teoria Tipurilor	Logică	Teoria categoriilor
tipuri	formule	obiecte
termeni	demonstraţii	săgeţi
inhabitation a tipului A	demonstraţie a lui A	săgeată $f : T \to A$
tip funcţie	implicaţie	exponenţi
tip produs	conjuncţie	produs
tip sumă	disjuncţie	coprodus
tipul void	false	obiect iniţial
tipul unit	true	obiect terminal$T$

Referinţe

- Roger Antonsen, *TED Talk: Math is the hidden secret to understanding the world*
  https://www.ted.com/talks/roger_antonsen_math_is_the_hidden_secret_to_understanding_the_world
- Samson Abramsky, *Categories, Proofs and Processed Lecture III - The Curry-Howard-Lambek Correspondence*
  http://www.math.helsinki.fi/logic/sellc-2010/course/LectureIII.pdf
- Philip Wadler, *Propositions as Types*
  https://homepages.inf.ed.ac.uk/wadler/papers/propositions-as-types/propositions-as-types.pdf
- Dan Grossman, *Lecture notes on The Curry-Howard Isomorphism*
  https://courses.cs.washington.edu/courses/cse505/12au/lec12_6up.pdf

Baftă la examen!