

Issue VI: Cartesian Interpreter

Maksym Sokhatskyi ¹

¹ National Technical University of Ukraine
Igor Sikorsky Kyiv Polytechnical Institute
2 червня 2025 р.

Анотація

Minimal language for sequential computations in cartesian closed categories.

Keywords: Lambda Calculus, Cartesian Closed Categories

Зміст

1	The Alonzo Language	2
1.1	Синтаксис	2
1.2	Правила обчислень	2
1.3	Підстановка	3
1.4	Рівність	3
1.5	Редукція	3
1.6	Нормалізація	3
1.7	Внутрішня мова ДЗК	3

1 The Alonzo Language

Мова програмування **Alonzo** — це чиста нетипізована мова, що є внутрішньою мовою декартово-замкнених категорій. Вона базується на лямбда-численні, розширеному парами, проєкціями та термінальним об'єктом, забезпечуючи мінімальну модель для обчислень у категорійному контексті.

1.1 Синтаксис

Терми **Alonzo** складаються зі змінних, лямбда-абстракцій, застосувань, пар, проєкцій (першої та другої) та термінального об'єкта. Це мінімальна мова, що підтримує обчислення через бета-редукцію та проєкції.

```
I = #identifier
O = I | ( O ) | O O | λ I → O | O , O | O.1 | O.2 | 1
```

```
type term =
| Var of string
| Lam of string * term
| App of term * term
| Pair of term * term
| Fst of term
| Snd of term
| Unit
```

1.2 Правила обчислень

Основними правилами обчислень у **Alonzo** є бета-редукція для лямбда-абстракцій та правила проєкцій для пар. Термінальний об'єкт є незвідним.

```
App (Lam (x, b), a) → subst x a b
Fst (Pair (t1, t2)) → t1
Snd (Pair (t1, t2)) → t2
```

$$\frac{(\lambda x.b) \ a \ \text{fst} \ \langle t_1, t_2 \rangle \ \text{snd} \ \langle t_1, t_2 \rangle}{b[a/x] \quad t_1 \quad t_2}$$

1.3 Підстановка

```
let rec subst x s = function
| Var y -> if x = y then s else Var y
| Lam (y, t) when x <> y -> Lam (y, subst x s t)
| App (f, a) -> App (subst x s f, subst x s a)
| Pair (t1, t2) -> Pair (subst x s t1, subst x s t2)
| Fst t -> Fst (subst x s t)
| Snd t -> Snd (subst x s t)
| Unit -> Unit
| t -> t
```

1.4 Рівність

```
let rec equal t1 t2 =
  match t1, t2 with
  | Var x, Var y -> x = y
  | Lam (x, b), Lam (y, b') -> equal b (subst y (Var x) b')
  | Lam (x, b), t -> equal b (App (t, Var x))
  | t, Lam (x, b) -> equal (App (t, Var x)) b
  | App (f1, a1), App (f2, a2) -> equal f1 f2 && equal a1 a2
  | Pair (t1, t2), Pair (t1', t2') -> equal t1 t1' && equal t2 t2'
  | Fst t, Fst t' -> equal t t'
  | Snd t, Snd t' -> equal t t'
  | Unit, Unit -> true
  | _ -> false
```

1.5 Редукція

```
let rec reduce = function
| App (Lam (x, b), a) -> subst x a b
| App (f, a) -> App (reduce f, reduce a)
| Pair (t1, t2) -> Pair (reduce t1, reduce t2)
| Fst (Pair (t1, t2)) -> t1
| Fst t -> Fst (reduce t)
| Snd (Pair (t1, t2)) -> t2
| Snd t -> Snd (reduce t)
| Unit -> Unit
| t -> t
```

1.6 Нормалізація

```
let rec normalize t =
  let t' = reduce t in
  if equal t t' then t else normalize t'
```

1.7 Внутрішня мова ДЗК

Мова **Alonzo** є внутрішньою мовою декартово-замкненої категорії (ДЗК). Вона включає лямбда-абстракції та застосування для замкнутої

структури, пари та проєкції для декартового добутку, а також термінальний об'єкт для відновлення повної структури ДЗК.

Література

- [1] Alonzo Church. *A Set of Postulates for the Foundation of Logic*. 1933.
- [2] Alonzo Church. *An Unsolvable Problem of Elementary Number Theory*. 1941.
- [3] Haskell Curry and Robert Feys. *Combinatory Logic, Volume I*. 1951.
- [4] Dana Scott. *A Type-Free Theory of Lambda Calculus*. 1970.
- [5] John C. Reynolds. *Towards a Theory of Type Structure*. 1974.
- [6] Henk Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. 1984.
- [7] G. Cousineau, P.-L. Curien, and M. Mauny. *The Categorical Abstract Machine*. 1985.