Безтипове лямбда-числення

1. Словник лямбда-числення

1. чотири спеціальні знаки 'λ', '. ', '(', ')';
2. злічення множина **Var**, елементи якої називають змінними: $v_0, v_1, \ldots, v_n, \ldots$, є атомарними конструктами числення.

Цей словник програмно можна реалізувати так

```python
class Var(int):

    def __new__(cls, x : Any) -> Self:
        try:
            ix = int(x)
        except ValueError:
            raise ValueError("Var() error! Bad argument")
        if ix < 0:
            raise ValueError("Var() error! Bad argument")
        return super().__new__(cls, ix)

    def __str__(self) -> str:
        return f"v{str(self)}"

    def __eq__(self, another: Self) -> bool:
        # Leibnitz's equality of variables
        if type(another) != Var:
            return False
        return super().__eq__(another)
```

2. Синтаксичні правила лямбда-числення

Вирази лямбда-числення називають лямбда-термами або просто термами. Терми будуються за допомогою наступних трьох правил

$$\frac{x : \mathbf{Var}}{x : \mathbf{Term}} \tag{1}$$

$$\frac{M : \mathbf{Term} \quad N : \mathbf{Term}}{(M\,N) : \mathbf{Term}} \tag{2}$$

$$\frac{x : \mathbf{Var} \quad M : \mathbf{Term}}{(\lambda\, x\, .\, M) : \mathbf{Term}} \tag{3}$$

Програмно ці правила можна реалізувати так:

```python
# Реалізуйте
```

```python
class Term(tuple):

# певний текст

    def __str__(self) -> str:
        pass

    pass
```

Будь-яка знакова конструкція, яка не може бути побудованою за допомогою цих трьох правил термом не є.

Приклад.

$\big(\lambda\, x\,.\; x\big)$ є термом:

1. x є змінною, а значить x також є термом за першим правилом;
2. $\big(\lambda\, x\,.\; x\big)$ за третім правилом.

Цей терм є моделлю наступної Python-програми

```python
def identity(x)
  return x
```

Синтаксис мови Python дозволяє цю функцію визначити і так

```python
identity = lambda x: x
```

Тоді терм $\Big(\big(\lambda\, x\,.\; x\big)\; y\Big)$ мовою Python можна представити так

```python
(lambda x: x)(y)
```

Натомість, терм $\Big(y\,\big(\lambda\, x\,.\; x\big)\Big)$ не має очевидної програмної інтерпретації.

Правила спрощення запису термів:

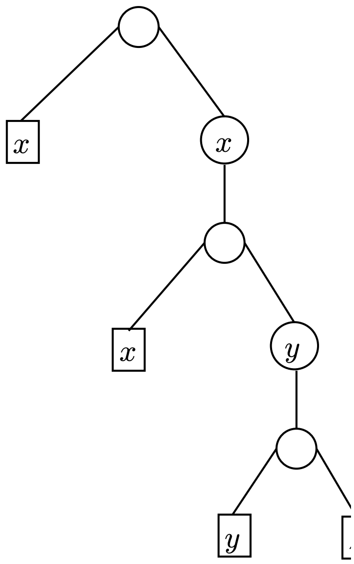
1. зовнішні дужки завжди опускають: $M\,N \equiv (M\,N)$, $\lambda\, x\,.\; M \equiv \big(\lambda\, x\,.\; M\big)$
2. застосування асоциативно ліворуч: $M\,N\,K \equiv ((M\,N)\,K)$
3. абстракція асоціативна праворуч: $\lambda\, x\,.\;\lambda\, y\,.\; M \equiv \Big(\lambda\, x\,.\big(\lambda\, y\,.\; M\big)\Big)$
4. застосування пріорітетніше за абстракцію: $\lambda\, x\,.\; M\,N \equiv \big(\lambda\, x\,.\,(M\,N)\big)$

Приклад.

$$x\,\lambda x\,.\;x\,\lambda y\,.\;y\,z \equiv \Big(x\Big(\lambda x\,.\Big(x\Big(\lambda y\,.\,(y\,z)\Big)\Big)\Big)\Big)$$

$$\xrightarrow{1}\; x\Big(\lambda x\,.\Big(x\Big(\lambda y\,.\,(y\,z)\Big)\Big)\Big) \xrightarrow{3} x\,\lambda x\,.\Big(x\Big(\lambda y\,.\,(y\,z)\Big)\Big) \xrightarrow{4} x\,\lambda x\,.\;x\Big(\lambda y\,.\,(y\,z)\Big)$$

$$\xrightarrow{3} x\,\lambda x\,.\;x\,\lambda y\,.\,(y\,z) \xrightarrow{4} x\,\lambda x\,.\;x\,\lambda y\,.\;y\,z$$



$$\Big(x\;\Big(\lambda x\,.\Big(x\;\Big(\lambda y\,.\,(y\;z)\Big)\Big)\Big)\Big) = x\;\lambda x\,.\;x\;\lambda y\,.\;y\;z$$

$$\text{'1010'} \mapsto (y\;z)$$

3. Правила перетворення термів (правила обчислення)

Практика символьних обчислень оперує з правилами

$$\begin{aligned}(a+b)^2 - a^2 &= \big(a^2+2ab+b^2\big) - a^2 \quad \text{використано правило } (x+y)^2 \to x^2+2xy+y^2\\ &= a^2+2ab+b^2-a^2 \quad \text{використано правило } \big(x^2+2xy+y^2\big) \to x^2+2xy+y^2\\ &= 2ab+b^2 \quad \text{використано правило } x+y-x \to y\end{aligned}$$

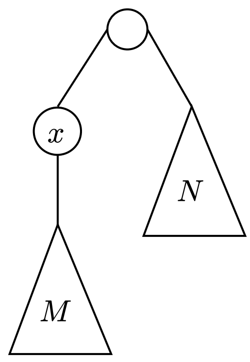
Як ми доводимо тотожність?

Ми застосовуємо правил перетворення виразів до лівої і правої частин тотожності з метою зведення кожної з них до спільної форми.

Саме такий процес Алонзо Черч використав для визначення поняття обчислення, формалізувавши його і назвавши β-редукцією.

3.1. Бета-редукція наївно. Вільні та зв'язані змінні

Терми виду $\big(\lambda\, x\, .\; M\big)N$ називаються бета-редексами (***red***ucible ***ex***presssion).



Правило β-редукції редекса $\big(\lambda\, x\, .\; M\big)N$

$$\big(\lambda\, x\, .\; M\big)N \underset{\beta}{\longrightarrow} M\big[x \leftarrow N\big],\ \text{де}$$

$M\big[x \leftarrow N\big]$ є результатом підстановки терму N в терм M замість всіх входжень змінної x в M .

Формально операція $M\big[x \leftarrow N\big]$ визначається так

- $x\big[x \leftarrow N\big] \equiv N$
- $y\big[x \leftarrow N\big] \equiv y$, якщо $y \neq x$
- $(M_1\, M_2)\big[x \leftarrow N\big] \equiv \Big(M_1\big[x \leftarrow N\big]\Big)\Big(M_2\big[x \leftarrow N\big]\Big)$
- $\big(\lambda\, y\, .\; M\big)\big[x \leftarrow N\big] \equiv \lambda\, y\, .\, \Big(M\big[x \leftarrow N\big]\Big)$

Наївний приклад.

$\big(\lambda\, x\, .\; x\big)\ M \underset{\beta}{\longrightarrow} x\big[x \leftarrow M\big] = M$

$\lambda\, x\, .\; x \equiv \mathtt{I}$ називається одиничним комбінатором

Аномалія

$F = \lambda\, x\,.\ \lambda\, y\,.\ yx$

$FMN \equiv \Big(\big(\lambda\, x\,.\ \lambda\, y\,.\ yx\big)M\Big)N \underset{\beta}{\rightarrow} \big(\lambda\, y\,.\ yM\big)N \underset{\beta}{\rightarrow} NM \Longrightarrow Fyx \underset{\beta}{\rightarrow} xy$

$Fyx \equiv \Big(\big(\lambda\, x\,.\ \lambda\, y\,.\ yx\big)y\Big)x \underset{\beta}{\rightarrow} \big(\lambda\, y\,.\ yy\big)x \underset{\beta}{\rightarrow} xx$

$xy\ =_\beta xx \Longrightarrow M\ =_\beta N$

$\lambda\, x\,.\ \lambda\, y\,.\ xy\ =_\beta \lambda\, x\,.\ \lambda\, y\,.\ xx$

$\Big(\big(\lambda\, x\,.\ \lambda\, y\,.\ xy\big)(\lambda\, z.\, z)\Big)M\ =_\beta \Big(\big(\lambda\, x\,.\ \lambda\, y\,.\ xx\big)(\lambda\, z.\, z)\Big)M$

$\Big(\big(\lambda\, x\,.\ \lambda\, y\,.\ xy\big)(\lambda\, z.\, z)\Big)M\ =_\beta \Big(\big(\lambda\, x\,.\ \lambda\, y\,.\ xx\big)(\lambda\, z.\, z)\Big)M$

$\Big(\big(\lambda\, x\,.\ \lambda\, y\,.\ xy\big)(\lambda\, z.\, z)\Big)M \underset{\beta}{\rightarrow} \big(\lambda\, y\,.\,(\lambda\, z.\, z)y\big)M \underset{\beta}{\rightarrow} \big(\lambda\, y\,.\ y\big)M \underset{\beta}{\rightarrow} M$

$\Big(\big(\lambda\, x\,.\ \lambda\, y\,.\ xx\big)(\lambda\, z.\, z)\Big)M \underset{\beta}{\rightarrow} \big(\lambda\, y\,.\,(\lambda\, z.\, z)(\lambda\, z.\, z)\big)M \underset{\beta}{\rightarrow} \big(\lambda\, y\,.\ \lambda\, z.\, z\big)M \underset{\beta}{\rightarrow} \lambda\, z.\, z$

3.2. Вільні та зв'язані змінні

3.2.1. Вільні та зв'язані змінні в математиці

Математичний приклад необхідності розрізняти вільні та зв'язані входження змінних:

$\sum_{k=1}^{n} k$ цей запис містить змінні n та k.

$$\begin{aligned}
\left(\sum_{k=1}^{n} k\right)\left[n \leftarrow 5\right] \quad &= \quad \sum_{k=1}^{5} k = 1+2+3+4+5 \\
\frac{n \cdot (n+1)}{2}\left[n \leftarrow 5\right] = \frac{5 \cdot 6}{2} = 15 \quad & \quad 1+2+3+4+5 = 15 \\
15 \quad &= \quad 15
\end{aligned}$$

коректно, бо змінна n є вільною.

$$\left(\sum_{k=1}^{n} k\right)\left[k \leftarrow 5\right] \stackrel{?}{=} \sum_{5=1}^{n} 5 \text{ ???}$$

некоректно, бо змінна k є зв'язаною.

Програмна ілюстрація:

```
def sum():
    nonlocal n
    acc = 0
    for k in range(1, n + 1):
        acc += k
    return acc
```

$$\int_{x_0}^{x} f(x)dx = \int_{x_0}^{x} f(y)dy$$

$$\left.\int_{x_0}^{x} f(x)dx\right|_{x \to 1} = \int_{x_0}^{1} f(y)dy$$

$$\left.\int_{x_0}^{x} f(x)dx\right|_{x \to 2} = \int_{x_0}^{2} f(y)dy$$

3.2.2. Вільні та зв'язані змінні в програмування

Приклад необхідності розрізняти вільні та зв'язані входження змінних у програмуванні:

```python
z = 10
def foo(x):
    def bar(y):
        return x + y
    return bar(x) + z

print(f"foo(5) = {foo(5)}")
# What is the expected output?
```

Цей код можна вважати ілюстрацією "терма" $\lambda\, x\,.\, \big(\lambda\, y\,.\; x + y\big) + z$.

Змінна z є вільною як у функціїї `foo`, так і в функції `bar`, а змінна x є також вільною у функції `bar`.

3.2.3. Вільні змінні формально

Вільні змінні лямбда-термів:

---

1. у термі $M \equiv x$ зміна x є вільною;
2. у термі $M\,N$ зміна x є вільною, якщо вона вільна в M або вона вільна в N;
3. у термі $\lambda\, y\,.\; M$ зміна x є вільною, якщо вона не співпадає з y та є вільною в M.

Множина вільних змінних терму M позначається через $\mathrm{FV}(M)$

$$\mathrm{FV}(M) = \begin{cases} \{x\} & \text{для } M \equiv x \\ \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2) & \text{для } M \equiv M_1\, M_2 \\ \mathrm{FV}(N) \setminus \{x\} & \text{для } M \equiv \lambda\, x\,.\; N \end{cases}$$

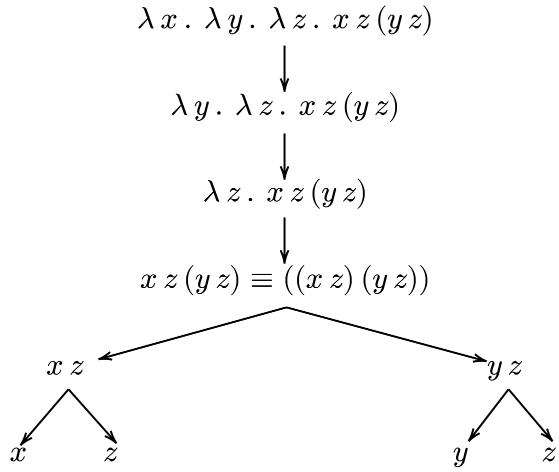
Терм, у якого немає вільних змінних, називається ***комбінатором***.

3.2.4. Підтерми

Множина $\mathrm{Sub}(M)$ підтермів терму M:

$$\mathrm{Sub}(M) = \begin{cases} \{x\} & \text{для } M \equiv x \\ \{M\} \cup \mathrm{Sub}(M_1) \cup \mathrm{Sub}(M_2) & \text{для } M \equiv M_1\, M_2 \\ \{M\} \cup \mathrm{Sub}(N) & \text{для } M \equiv \lambda\, x\,.\; N \end{cases}$$

Які підтерми є у терма $\lambda\, x\, .\ \lambda\, y\, .\ \lambda\, z\, .\ x\, z\, (y\, z)$?

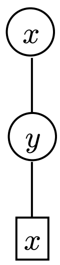


Підтерми: $\lambda\, x\, .\ \lambda\, y\, .\ \lambda\, z\, .\ x\, z\, (y\, z)$, $\lambda\, y\, .\ \lambda\, z\, .\ x\, z\, (y\, z)$, $\lambda\, z\, .\ x\, z\, (y\, z)$, $x\, z\, (y\, z)$, $x\, z$, $y\, z$, x, y, z

3.2.5. Зв'язані змінні формально

Змінна x є зв'язаною у термі M, якщо у M існує підтерм N виду $\lambda\, x\, .\ N'$.
У цьому випадку N' називають областю видимості (scope) для змінної x.

$\mathtt{K} = \lambda\, x\, .\ \lambda\, y\, .\ x$




```python
def K(x):
    def infun(y):
        return x
    return infun
```

$(\lambda\, x\, .\ \lambda\, y\, .\ x)y \xrightarrow[\beta]{} (\lambda\, y\, .\ x)[x \leftarrow y] = \lambda\, y\, .\ y$ - некоректна редукція редекса

$(\lambda\, x\, .\ \lambda\, y\, .\ x)y \xrightarrow[\alpha]{} (\lambda\, x\, .\ \lambda\, z\, .\ x)y \xrightarrow[\beta]{} (\lambda\, z\, .\ x)[x \leftarrow y] = \lambda\, z\, .\ y$

3.2.6. Альфа-конверсія

Змінну x називають ***свіжою*** для термів $M_1, \ldots, M_n$, якщо вона не фігурує в жодному з цих термів, тобто

1. випадок одного терма
   1.1. будь-яка змінна, що не співпадає з x, є свіжою для терма $M \equiv x$;
   1.2. будь-яка змінна, що є свіжою як для M_1, так і для M_2, є свіжою для $M \equiv M_1\, M_2$;
   1.3. будь-яка змінна, що є свіжою для N і не співпадає з x, є свіжою для $M \equiv \lambda\, x\,.\, N$.
2. будь-яка змінна, що є свіжою для кожного з термів $M_1, \ldots, M_n$, є свіжою для термів $M_1, \ldots, M_n$.

Альфа-конверсія - це перетворення переіменування зв'язаної змінної свіжою змінною

Якщо M є термом, x його зв'язаною змінною, а y - змінною свіжою для M, тоді результатом заміни x на y в M (позначається через $M[y\,/\,x]$) називають терм

$$M[y\,/\,x] = \begin{cases} z & \text{для } M \equiv z \\ (M_1[y\,/\,x])\,(M_2[y\,/\,x]) & \text{для } M \equiv M_1\, M_2 \\ \lambda\, z\,.\,(N[y\,/\,x]) & \text{для } M \equiv \lambda\, z\,.\, N,\ z \neq x \\ \lambda\, y\,.\,\big(N\big[x \leftarrow y\big]\big) & \text{для } M \equiv \lambda\, x\,.\, N \text{ і всіх вільних входжень } x \text{ в } N \end{cases}$$

таке перетворення позначається через $M \xrightarrow[\alpha]{} N$.
Відношення альфа-конверсії $M \xrightarrow[\alpha]{*} N$ є скінченною (в тому числі порожньою) низкою альфа-конверсій, які ведуть від M до N.

Відношення альфа-конгруентністі

Терми M та N є альфа конгруентними $M =_\alpha N$, якщо для якогось терму K вірно як $M \xrightarrow[\alpha]{*} K$, так і $N \xrightarrow[\alpha]{*} K$.
Це відношення є розв'язним відношенням еквівалентності.

Справедливість $M =_\alpha M$ є очевидною, оскільки як можна взяти $K = M$ і порожню низкою альфа-конверсій.
Справедливість $M =_\alpha N \Longrightarrow N =_\alpha M$ є також очевидною.
Припустимо тепер, що $M_1 =_\alpha M_2$ та $M_2 =_\alpha M_3$, тобто маємо $M_1 \xrightarrow[\alpha]{*} K_{12}$, $M_2 \xrightarrow[\alpha]{*} K_{12}$, $M_2 \xrightarrow[\alpha]{*} K_{23}$ та $M_3 \xrightarrow{*} K_{23}$. Оскільки як K_{12}, так і K_{23} є результатами альфа-конверсії терма M_2, то вони відрізняються тільки зв'язаними змінними, а тому заміняючи ці змінні свіжими отримаємо $K_{12} \xrightarrow[\alpha]{*} K$ та $K_{23} \xrightarrow[\alpha]{*} K$ для деякого терму K.
Розв'язність встановлюється індукцією за структурою терма. □

Принцип Х. Берендрехта

Терм M і всі терми, що є α-конгруентними йому не розрізняються.

3.3. Уточнена бета-редукція

Проблема

$(\lambda\, x\, .\, M)\, N \underset{\beta}{\longrightarrow} M[x := N]$

$(\lambda\, x\, .\, \lambda\, y\, .\, x)\, y \overset{?}{\underset{\beta}{\longrightarrow}} \lambda\, y\, .\, y$

Бета-редукцією редекса $(\lambda\, x\, .\, M)N$ називають терм, що отриманий шляхом підстановки терма N замість всіх вільних входжень змінної x в терм M' $\left(M'\left[x \leftarrow N\right]\right)$, де M' є такою α-конверсією M, що всі вільні в N змінні не є зв'язаними в M'.
1-бета-редукцією терма M до терма N $\left(M \underset{\beta}{\longrightarrow} N\right)$ є процес заміни редекса в M на результат його бета-редукції, в наслідок чого терм M перетворюється на терм N.
Бета-редукцією терма M до терма N $\left(M \overset{*}{\underset{\beta}{\longrightarrow}} N\right)$ є низка (в тому числі порожня) 1-бета-редукцій, що перетворюють терм M на N.

$(\lambda\, x\, .\, \lambda\, y\, .\, x)\, y \underset{\alpha}{\longrightarrow} (\lambda\, x\, .\, \lambda\, z\, .\, x)\, y \underset{\beta}{\longrightarrow} \lambda\, z\, .\, y \underset{\alpha}{\longrightarrow} \lambda\, x\, .\, y$

$F \equiv \lambda\ x\ .\ \lambda\ y\ .\ yx$

$Fyx \equiv \big((\lambda\ x\ .\ \lambda\ y\ .\ yx)y\big)x \underset{\alpha}{\longrightarrow} \big((\lambda\ x\ .\ \lambda\ z\ .\ zx)y\big)x \underset{\beta}{\longrightarrow} (\lambda\ z\ .\ zy)x \to xy$

Бета-редукція необов'язково веде до термів, які не мають редексів і, тому вже не можуть перетворюватися далі.

Приклади, коли бета-редукція не завершується

Петля редукції:

$$(\lambda x\,.\,x\,x)\lambda x\,.\,x\,x \underset{\beta}{\to} (x\,x)\big[x := (\lambda x\,.\,x\,x)\big] \equiv \big((\lambda x\,.\,x\,x)\;(\lambda x\,.\,x\,x)\big)$$
$$\equiv (\lambda x\,.\,x\,x)\lambda x\,.\,x\,x$$

Нескінченна послідовність редукцій без повторень:

$$(\lambda x\,.\,x\,x\,x)\lambda x\,.\,x\,x\,x \underset{\beta}{\to} ((x\,x)\,x)\big[x \leftarrow (\lambda x\,.\,x\,x\,x)\big]$$
$$\equiv \big((\lambda x\,.\,x\,x\,x)(\lambda x\,.\,x\,x\,x)\big)\lambda x\,.\,x\,x\,x$$
$$\underset{\beta}{\to} \Big(\big((\lambda x\,.\,x\,x\,x)(\lambda x\,.\,x\,x\,x)\big)(\lambda x\,.\,x\,x\,x)\Big)\lambda x\,.\,x\,x\,x\ldots$$

Неоднозначність

$$\underbrace{(\lambda x\,.\,\lambda y\,.\,y)\overbrace{\big((\lambda x\,.\,x\,x\,x)\lambda x\,.\,x\,x\,x\big)}^{\text{int-redex}}}_{\text{ext-redex}} \underset{\beta}{\to} \lambda y\,.\,y \quad \text{при редукції ext-redex}$$

$$\underset{\beta}{\to} (\lambda x\,.\,\lambda y\,.\,y)\Big(\big((\lambda x\,.\,x\,x\,x)\lambda x\,.\,x\,x\,x\big)\lambda x\,.\,x\,x\,x\Big)$$

3.4. Ета-конверсія

Візьмемо терм M та змінну $x \notin \mathrm{FV}(M)$ і утворимо терм $\lambda x\,.\,M\,x$.

Для довільного терму N маємо $(\lambda x\,.\,M\,x)N \underset{\beta}{\to} (Mx)[x := N] \equiv MN$. Тобто, з точки зору застосувань (аплікативної точки зору) терми $(\lambda x\,.\,M\,x)$ та M не розрізняються, що призводить до необхідності їх ототожнити у разі, якщо ми вважаємо терми функціями. На рівні відношення $\underset{\beta}{\to}$ це зробити неможливо, як показують терми x та $\lambda y\,.\,xy$, які обидва є нормальними формами. Тому необхідне додаткове правило ета-конверсії

$$\lambda x\,.\,M\,x \underset{\eta}{\to} M, \quad \text{якщо } x \notin \mathrm{FV}(M) \tag{4}$$

3.5. Обчислення як бете-ета-редукція

Об'єднуючи всі ци правила можемо визначити відношення бета-ета-редукції:

$$\begin{aligned} M \xrightarrow[\beta\eta]{*} N \Leftrightarrow \quad & M = N \vee \\ & \Big(M \underset{\alpha}{\to} M' \wedge M' \xrightarrow[\beta\eta]{*} N\Big) \vee \\ & \Big(M \underset{\beta}{\to} M' \wedge M' \xrightarrow[\beta\eta]{*} N\Big) \vee \\ & \Big(M \underset{\eta}{\to} M' \wedge M' \xrightarrow[\beta\eta]{*} N\Big). \end{aligned} \tag{5}$$