Sistema de Transformação de Programas

Miguel Gonçalves

Programação Funcional Avançada

O sistema MAG

- Utilização da lei da fusão
- Optimização de código
- Programador: especificação da função a optimizar

Lei da Fusão

- -definição indutiva do tipo lista

$$\text{data Lista a} = \text{Vazia} \mid (a, Lista\ a)$$

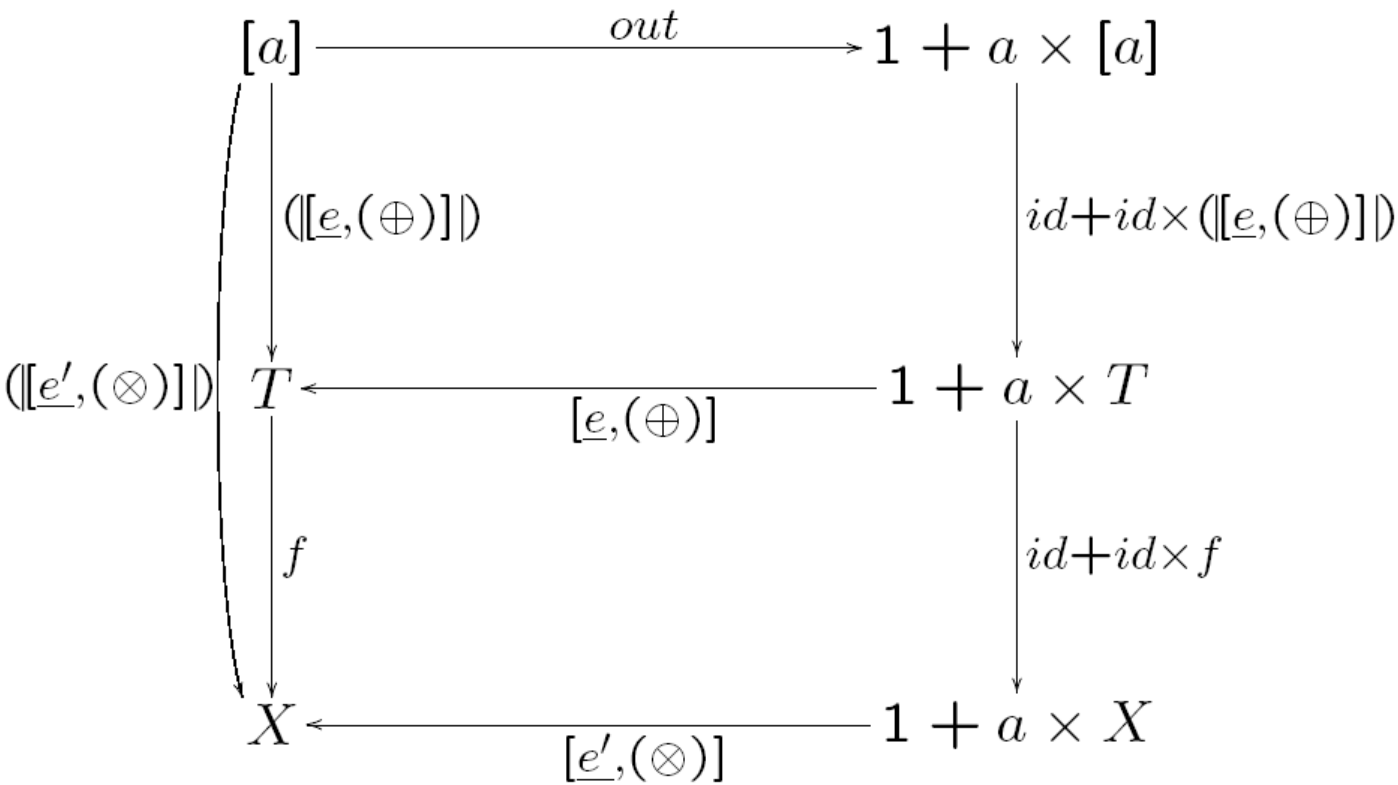
- -lei da fusão para listas

$$\text{f (foldr } (\oplus)\ e\ xs) = foldr\ (\otimes)\ e'\ xs$$

$$\text{if f e} = \text{e'}$$

$$\forall x, y \cdot f(x \oplus y) = x \otimes (f\, y)$$

Lei da Fusão



Lei da Fusão

$$\mathrm{f}\ [\mathrm{e},(\oplus)] = [e', (\otimes)](id + id \times f)$$

$$\iff \{fusão - +;\ absorção - +\}$$
$$[\mathrm{f}\ \mathrm{e},\mathrm{f}\ (\oplus)] = [e'\ id, (\otimes)(id \times f)]$$

$$\iff \{igualdade\ estrutural\ either;\ nat - id\}$$
$$\mathrm{f}\ \mathrm{e} = \mathrm{e'}\ \wedge\ f\ (\oplus) = (\otimes)(id \times f)$$

$$\iff \{introdução\ da\ variável\ (x, y)\}$$
$$\mathrm{f}\ \mathrm{e} = \mathrm{e'}\ \wedge\ f\ (\oplus)(x, y) = (\otimes)(id \times f)(x, y)$$

$$\iff \{definição\ de\ \times;\ operador\ infixo;\ nat - id\}$$
$$\mathrm{f}\ \mathrm{e} = \mathrm{e'}\ \wedge\ f(x \oplus y) = (\otimes)(x, f\ y)$$

$$\iff \{operador\ infixo\}$$
$$\mathrm{f}\ \mathrm{e} = \mathrm{e'}\ \wedge\ f\ (x \oplus y) = x \otimes (f\ y)$$

Reverse

- Algoritmo quadrático
- Recursividade “clássica”

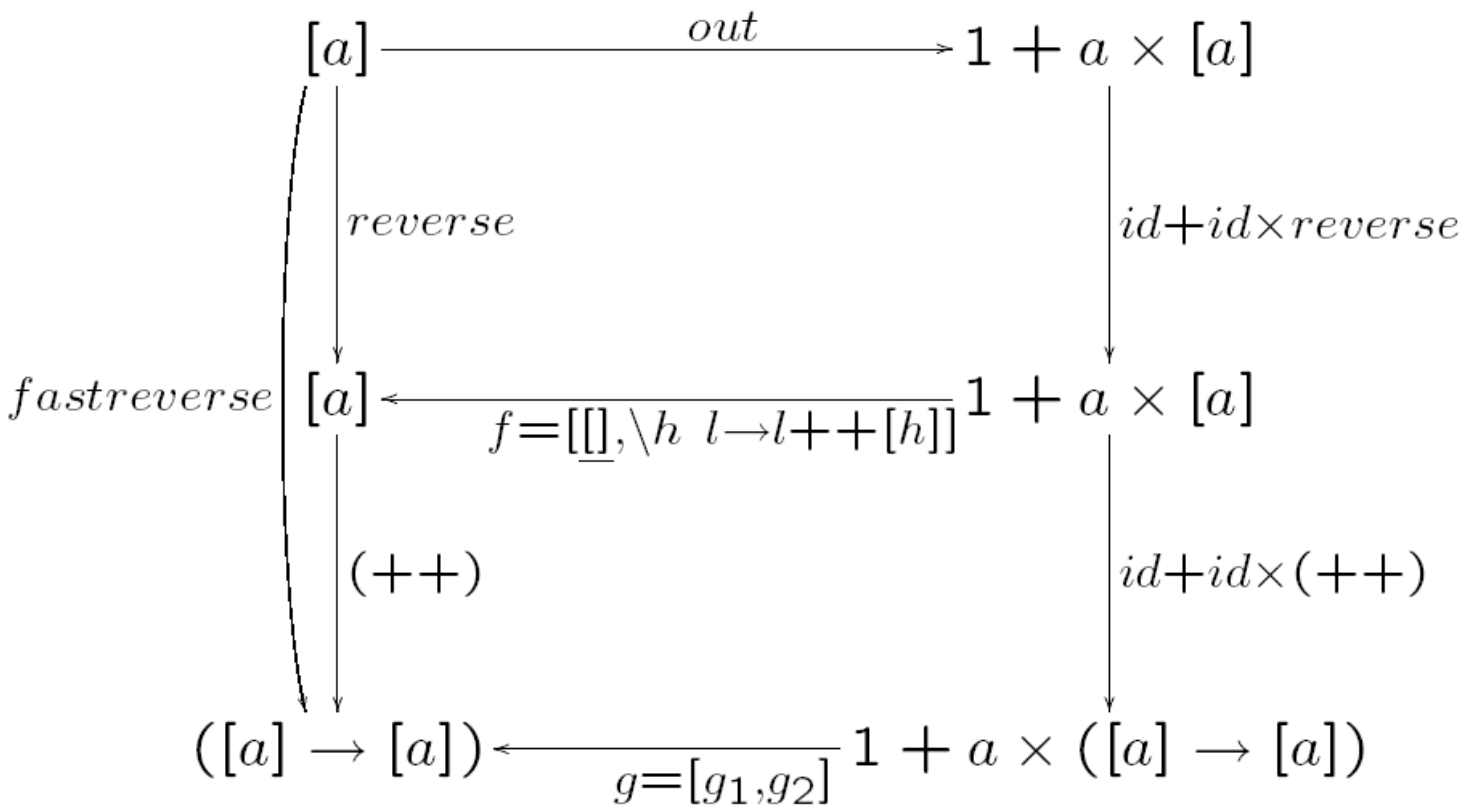
```
--reverse com recursividade "clássica"

reverse_dummy [] = []
reverse_dummy (h:t) = reverse_dummy t ++ [h]
```

Reverse

- Especificação: uso de acumuladores

$$
\begin{aligned}
& \text{fastreverse xs ys} = \text{reverse xs} +\!+ \text{ ys} \\
\iff & \\
& \text{fastreverse xs ys} = (+\!+)(\text{reverse xs})\ \text{ys} \\
\iff & \\
& \text{fastreverse xs} = (+\!+)(\text{reverse xs}) \\
\iff & \\
& \text{fastreverse} = (+\!+).\text{reverse}
\end{aligned}
$$

Reverse


[a]
out
1 + a × [a]
reverse
id+id×reverse
fastreverse
[a]
f=[[],\h l→l++[h]]
1 + a × [a]
(++)
id+id×(++)
([a] → [a])
g=[g1,g2]
1 + a × ([a] → [a])