

$$1) C[(y x)] = \lambda x. \lambda y. x (y \lambda x. (y x))$$

$$= \lambda x. \lambda y. x (y \lambda x_1 (y x_1)) \checkmark$$

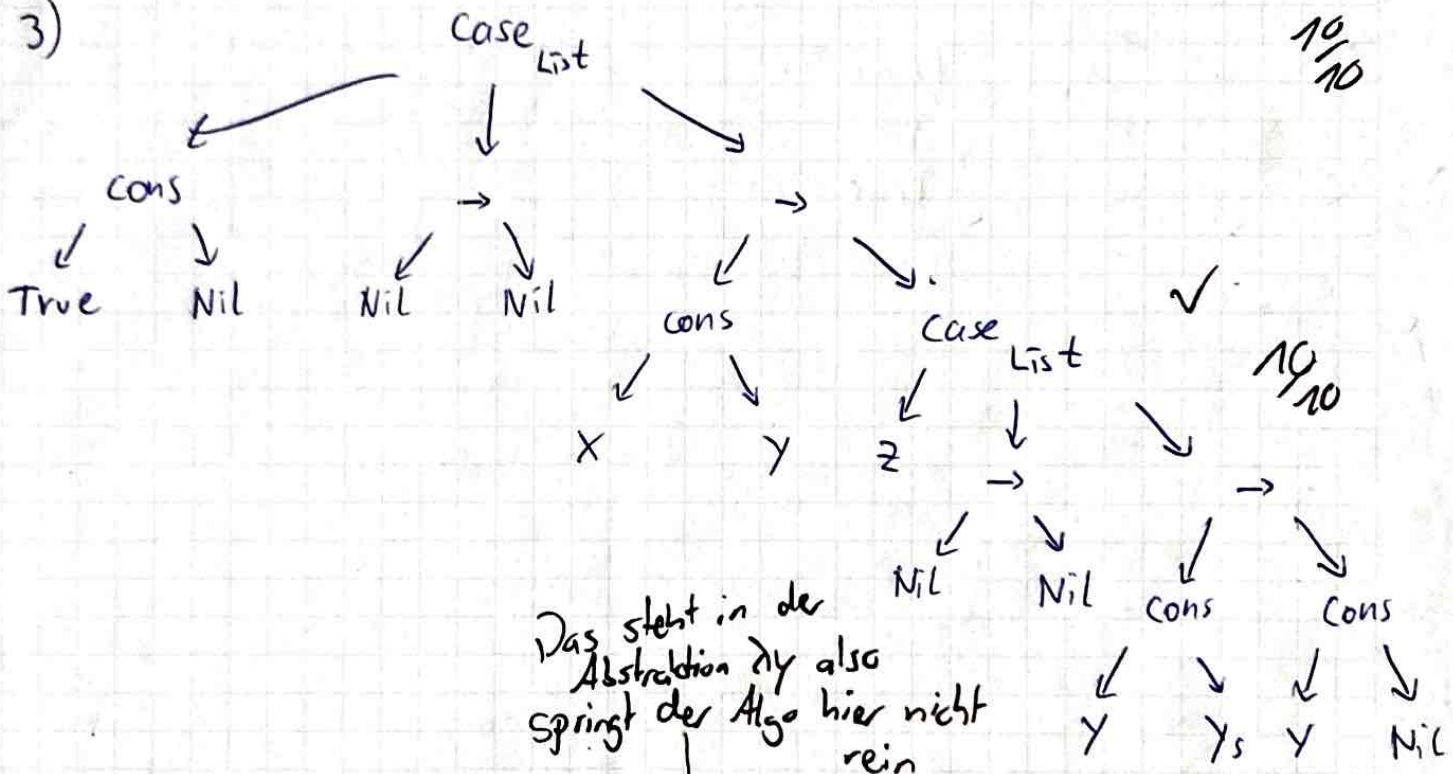
$$C[D[\frac{z}{x}]] = \lambda x. \lambda y. x (y \lambda x_1 (\lambda z. z)) \checkmark$$

~~$\lambda x. \lambda y. x (y \lambda x. (y x))$~~

$$D[C[\lambda x. (x y)]] = \lambda z. \lambda x. \lambda y. x (y \lambda x. \lambda x. (x y))$$

$$= \lambda z. \lambda x. \lambda y. x (y \lambda x_1. (\lambda x_2. (x_2 y))) \checkmark$$

3)



$$2b) ((\lambda x. \lambda y. y (y x)) ((\lambda z. (\lambda u. u z) (\lambda w. w))) (\lambda v. v))$$

$$\xrightarrow{\alpha_0} ((\lambda y. y (y ((\lambda z. (\lambda u. u z) (\lambda w. w)))) (\lambda v. v))$$

$$\xrightarrow{\alpha_0} ((\lambda y. y (y ((\lambda u. u (\lambda w. w)))) (\lambda v. v))$$

$$\xrightarrow{\alpha_0} \lambda v. v (\lambda v. v (\lambda u. u (\lambda w. w))) \xrightarrow{\alpha_0} \lambda u. u (\lambda w. w) \xrightarrow{\alpha_0} \lambda w. w$$

5/10

Die Klammer ist wichtig

$$c) ((\lambda x. \lambda y. y (y x)) ((\lambda z. (\lambda u. u z) (\lambda w. w))) (\lambda v. v))$$

$$\xrightarrow{\alpha_0} ((\lambda x. \lambda y. y (y x)) ((\lambda u. u (\lambda w. w))) (\lambda v. v))$$

$$\xrightarrow{\alpha_0} ((\lambda x. \lambda y. y (y x)) (\lambda w. w))$$

$$\xrightarrow{\alpha_0} ((\lambda y. y (y (\lambda w. w))) \xrightarrow{\alpha_0} \lambda w. w$$

Dies steht in \lambda u hier springt er nicht rein

falscher Typ  
 ↓  
 case<sub>List</sub> True ...  
 case<sub>Typ</sub> (λy. y) ...

Abstraktion

True True  
 ↑  
 überflüssiger Konstruktor

True  
 |  
 True

Cons (True True) Nil  
 / \  
 True Nil  
 |  
 True

Normal ( ) ←

Anwendung

App

To-Do  
 2F  
 Case. Typisierung  
 Iterativ, Miller  
 Produkt Konstante