

Induktionsanfang XS= Nil: Sei e nd ys beliebig.

LS = Nil @ (Conse e ys) = cons e ys RS= (mrc Nil e) A ys=(cons e Nil) A ys +) Com e (Nil @ ys) = Con e ys

Sei XS eine List und

Induktionsschritt:

Zu reigen: $\forall x$ gilt: $A(Cons \times xs)$ Sei e van Typ a ud ys van Typ Reit a bebrig.

2n reigh: (Cons x xs) ⊕ (Cons e ys) = (moc (Cons x xs)

->LS=RSV

1 snoc Nil y = Cons y Nil 2 snoc (Cons x xs) y = Cons x (snoc xs y) 3 Nil \oplus ys = ys + (Cons x xs) \oplus ys = Cons x (xs \oplus ys)

data List a = Nil | Cons a (List a)

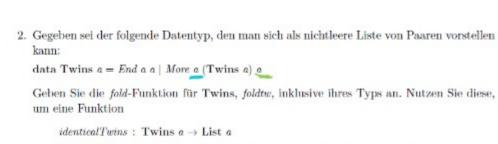
Beweisen Sie mittels struktureller Induktion, dass

 $\forall e, xs, ys. xs \oplus (Cons \ e \ ys) = (snoc \ xs \ e) \oplus ys.$

Geben Sie im Induktionsschritt die Induktionsannahme explizit an und erläutern Sie alle

$$\int_{V} \int_{S} = Cons \times ((mre \times S \in) \oplus ys) =$$

$$= ((cons \times (mre \times S \in)) \oplus ys)$$



zu definieren, die Elemente nur dann in die Zielliste übernimmt, wenn die beiden Werte übereinstimmen. Beispielsweise soll gelten: Startakkumulator

identicalTwins (More 1 (More 2 (More 1 (End 2 0) 1) 2) 3) = [2,1]

Hinweis: Sie dürfen die üblichen Konstrukte wie λ -Abstraktion, if . . . then λ -else verwenden und annehmen, dass Elemente mittels == auf Gleichheit überprüß werden können.

akk > if x==y then Cans x akk doe akk)

data List
$$a = Nil \mid Cons \ a \ (List \ a)$$

Lann man weglanen

miro