Contents

1 exType Theory

Built: 06 February 2020
Parent Theories: indexedLists, patternMatches

1.1 Definitions

[APP_def]

$\vdash$ (∀ l. APP [] l = l) ∧ ∀ h l_1 l_2. APP (h::l_1) l_2 = h::APP l_1 l_2

[Map_def]

$\vdash$ (∀ f. Map f [] = []) ∧ ∀ f h l_1. Map f (h::l_1) = f h::Map f l_1

1.2 Theorems

[APP_ASSOC]

$\vdash$ ∀ l_1 l_2 l_3. APP (APP l_1 l_2) l_3 = APP l_1 (APP l_2 l_3)

[LENGTH_APP]

$\vdash$ ∀ l_1 l_2. LENGTH (APP l_1 l_2) = LENGTH l_1 + LENGTH l_2

[Map_APP]

$\vdash$ Map f (APP l_1 l_2) = APP (Map f l_1) (Map f l_2)

2 nexp Theory

Built: 06 February 2020
Parent Theories: indexedLists, patternMatches

2.1 Datatypes

nexp = Num num | Add nexp nexp | Sub nexp nexp | Mult nexp nexp

2.2 Definitions

[nexpVal_def]

$\vdash$ (∀ *num*. nexpVal (Num *num*) = *num*) ∧
(∀ f_1 f_2. nexpVal (Add f_1 f_2) = nexpVal f_1 + nexpVal f_2) ∧
(∀ f_1 f_2. nexpVal (Sub f_1 f_2) = nexpVal f_1 - nexpVal f_2) ∧
∀ f_1 f_2. nexpVal (Mult f_1 f_2) = nexpVal f_1 × nexpVal f_2

2.3 Theorems

[Add_0]

$\vdash \forall f.$ nexpVal (Add (Num 0) f) = nexpVal f

[Add_SYM]

$\vdash \forall f_1\ f_2.$ nexpVal (Add f_1 f_2) = nexpVal (Add f_2 f_1)

[Mult_ASSOC]

$\vdash \forall f_1\ f_2\ f_3.$
nexpVal (Mult f_1 (Mult f_2 f_3)) =
nexpVal (Mult (Mult f_1 f_2) f_3)

[Sub_0]

$\vdash \forall f.$
(nexpVal (Sub (Num 0) f) = 0) $\wedge$
(nexpVal (Sub f (Num 0)) = nexpVal f)

Index