

```

(*****
(* ProgrammingLanguageReification *)
(* *)
(* Reifications in ProgrammingLanguages. *)
(* *)
(*****)

```

Generalizable All Variables.

```

Require Import Preamble.
Require Import General.
Require Import Categories_ch1_3.
Require Import InitialTerminal_ch2_2.
Require Import Functors_ch1_4.
Require Import Isomorphisms_ch1_5.
Require Import ProductCategories_ch1_6_1.
Require Import OppositeCategories_ch1_6_2.
Require Import Enrichment_ch2_8.
Require Import Subcategories_ch7_1.
Require Import NaturalTransformations_ch7_4.
Require Import NaturalIsomorphisms_ch7_5.
Require Import MonoidalCategories_ch7_8.
Require Import Coherence_ch7_8.
Require Import Enrichment_ch2_8.
Require Import RepresentableStructure_ch7_2.
Require Import FunctorCategories_ch7_7.

```

```

Require Import Reification.
Require Import NaturalDeduction.
Require Import NaturalDeductionCategory.
Require Import ProgrammingLanguage.
Require Import Enrichments.

```

Section ProgrammingLanguageReification.

```

Definition TwoLevelLanguage '(Guest:ProgrammingLanguage) '(Host:ProgrammingLanguage)
  := Reification (TypesEnrichedInJudgments Guest) (TypesEnrichedInJudgments Host) [].

```

```

Inductive NLevelLanguage : forall (n:nat) '(PL:ProgrammingLanguage), Type :=
| NLevelLanguage_zero : forall '(lang:ProgrammingLanguage),
  NLevelLanguage 0 lang
| NLevelLanguage_succ : forall '(L1:ProgrammingLanguage) '(L2:ProgrammingLanguage) n,

```

TwoLevelLanguage L1 L2 -> NLevelLanguage n L1 -> NLevelLanguage (S n) L2.

```
(*
Definition OmegaLevelLanguage : Type :=
  { f : nat -> ProgrammingLanguage
    & forall n, TwoLevelLanguage (f n) (f (S n)) }.
*)
```

End ProgrammingLanguageReification.

```
(*
Structure ProgrammingLanguage :=
{ plsmme_t      : Type
; plsmme_judg  : Type
; plsmme_sequent : Tree ??plsmme_t -> Tree ??plsmme_t -> plsmme_judg
; plsmme_rule   : Tree ??plsmme_judg -> Tree ??plsmme_judg -> Type
; plsmme_pl     : @ProgrammingLanguage plsmme_t plsmme_judg plsmme_sequent plsmme_rule
; plsmme_smme   : SurjectiveEnrichment (TypesEnrichedInJudgments _ _ plsmme_pl)
}.
Coercion plsmme_pl : ProgrammingLanguage >-> ProgrammingLanguage.
Coercion plsmme_smme : ProgrammingLanguage >-> SurjectiveMonicMonoidalEnrichment.
*)
```