Enriched Category Basics

Eugene W. Stark

Department of Computer Science
Stony Brook University
Stony Brook, New York 11794 USA

March 17, 2025

Abstract

The notion of an enriched category generalizes the concept of category by replacing the hom-sets of an ordinary category by objects of an arbitrary monoidal category. In this article we give a formal definition of enriched categories and we give formal proofs of a relatively narrow selection of facts about them. One of the main results is a proof that a closed monoidal category can be regarded as a category "enriched in itself". The other main result is a proof of a version of the Yoneda Lemma for enriched categories.

Contents

Introduction

The notion of an enriched category [1] generalizes the concept of category by replacing the hom-sets of an ordinary category by objects of an arbitrary monoidal category $\mathcal{V}$. The choice, for each object a, of a distinguished element $id\ a : a \rightarrow a$ as an identity, is replaced by an arrow $Id\ a : \mathcal{I} \rightarrow Hom\ a\ a$ of $\mathcal{V}$. The composition operation is similarly replaced by a family of arrows $Comp\ a\ b\ c : Hom\ B\ C \otimes Hom\ A\ B \rightarrow Hom\ A\ C$ of $\mathcal{V}$. The identity and composition are required to satisfy unit and associativity laws which are expressed as commutative diagrams in $\mathcal{V}$. Of particular interest is the case in which $\mathcal{V}$ is symmetric monoidal and closed; in that case, as Kelly states ([1], Section 1.6): "The structure of $\mathcal{V}$-**CAT** then becomes rich enough to permit of Yoneda-lemma arguments formally identical with those in **CAT**."

The goal of this article is to formalize the basic definition of enriched category and some related notions, and to prove a relatively narrow selection of facts about these definitions. For reference and inspiration, we follow the early sections of the book by Kelly [1]; however a comprehensive formalization of the material in that book is explicitly not our objective here. Rather, beyond the basic definitions we are primarily interested in the following two results: (1) that a closed monoidal category can be regarded as a category "enriched in itself"; and (2) the Yoneda Lemma for enriched categories (specifically, the weak form considered in Section 1.9 of [1]). We needed the basic definitions and result (1) for use in a separate article [4]. Although this material could have been included as part of that other article, as it is general material that does not depend on the specific application considered there, it seemed best to present it as a stand-alone development that would be more readily accessible for use by others. As far as result (2) is concerned, we originally formalized and proved it as part of our exploration leading up to [4]. Ultimately, we did not find result (2) to be necessary for the satisfactory development of that work, but as it is a result of general interest whose formalization did involve some struggle to achieve, it seems worthwhile to include it here.

This article is organized as follows: In Chapter 1 we give formal definitions for the notions "closed monoidal category" and "cartesian closed monoidal category" and prove some facts about them. This builds on the formal development of the theory of monoidal categories in our previous article [3]. The main goals of this section are to prove some general facts about exponentials that are used in [4], and to do most of the preliminary work (the parts that do not specifically depend on the definition of enriched category) involved in showing that a closed monoidal category is "enriched in itself". In Chapter 2 we give definitions for "enriched category" and the related notions "enriched functor," "enriched natural transformation," and "underlying category," and we complete the formal statement and proof of "self-enrichment." We then continue with the definition of the opposite of an enriched category, give definitions for the notions of covariant and contravariant enriched hom functors, and prove corresponding covariant and contravariant versions of the Yoneda Lemma.

Chapter 1

Closed Monoidal Categories

A *closed monoidal category* is a monoidal category such that for every object b, the functor - $\otimes$ b is a left adjoint functor. A right adjoint to this functor takes each object c to the *exponential* $exp\ b\ c$. The adjunction yields a natural bijection between $hom\ (- \otimes b)\ c$ and $hom\ -\ (exp\ b\ c)$. In enriched category theory, the notion of "hom-set" from classical category theory is generalized to that of "hom-object" in a monoidal category. When the monoidal category in question is closed, much of the theory of set-based categories can be reproduced in the more general enriched setting. The main purpose of this section is to prepare the way for such a development; in particular we do the main work required to show that a closed monoidal category is "enriched in itself."

theory *ClosedMonoidalCategory*
imports *MonoidalCategory.CartesianMonoidalCategory*
begin

1.1 Definition and Basic Facts

As is pointed out in [2], unless symmetry is assumed as part of the definition, there are in fact two notions of closed monoidal category: *left*-closed monoidal category and *right*-closed monoidal category. Here we define versions with and without symmetry, so that we can identify the places where symmetry is actually required.

locale *closed-monoidal-category* =
monoidal-category +
assumes *left-adjoint-tensor*: $\bigwedge b.\ ide\ b \Longrightarrow left\text{-}adjoint\text{-}functor\ C\ C\ (\lambda x.\ x \otimes b)$

locale *closed-symmetric-monoidal-category* =
closed-monoidal-category +
symmetric-monoidal-category

Similarly to what we have done in previous work, besides the definition of *closed-monoidal-category*, which adds an assumed property to *monoidal-category* but not any additional structure, we find it convenient also to define *elementary-closed-monoidal-category*, which assumes particular exponential structure to have been chosen, and uses this given structure to express the properties of a closed monoidal category in a more elementary way.

locale *elementary-closed-monoidal-category* =
  monoidal-category +
fixes *exp* :: $'a \Rightarrow 'a \Rightarrow 'a$
and *eval* :: $'a \Rightarrow 'a \Rightarrow 'a$
and *Curry* :: $'a \Rightarrow 'a \Rightarrow 'a \Rightarrow 'a \Rightarrow 'a$
assumes *eval-in-hom-ax*: ⟦ *ide b*; *ide c* ⟧ ⟹ «*eval b c* : *exp b c* ⊗ *b* → *c*»
and *ide-exp* [*intro*, *simp*]: ⟦ *ide b*; *ide c* ⟧ ⟹ *ide* (*exp b c*)
and *Curry-in-hom-ax*: ⟦ *ide a*; *ide b*; *ide c*; «*g* : *a* ⊗ *b* → *c*» ⟧
  ⟹ «*Curry a b c g* : *a* → *exp b c*»
and *Uncurry-Curry*: ⟦ *ide a*; *ide b*; *ide c*; «*g* : *a* ⊗ *b* → *c*» ⟧
  ⟹ *eval b c* · (*Curry a b c g* ⊗ *b*) = *g*
and *Curry-Uncurry*: ⟦ *ide a*; *ide b*; *ide c*; «*h* : *a* → *exp b c*» ⟧
  ⟹ *Curry a b c* (*eval b c* · (*h* ⊗ *b*)) = *h*

locale *elementary-closed-symmetric-monoidal-category* =
  symmetric-monoidal-category +
  elementary-closed-monoidal-category
begin

  sublocale *elementary-symmetric-monoidal-category*
    C tensor $\mathcal{I}$ *lunit runit assoc sym*
    ⟨*proof*⟩

end

We now show that, except for the fact that a particular choice of structure has been made, closed monoidal categories and elementary closed monoidal categories amount to the same thing.

1.1.1 An ECMC is a CMC

context *elementary-closed-monoidal-category*
begin

  notation *Curry* (‹*Curry*[-, -, -]›)

  abbreviation *Uncurry* (‹*Uncurry*[-, -]›)
  where *Uncurry*[*b*, *c*] *f* ≡ *eval b c* · (*f* ⊗ *b*)

  lemma *Curry-in-hom* [*intro*]:
  assumes *ide a* **and** *ide b* **and** «*g* : *a* ⊗ *b* → *c*» **and** *y* = *exp b c*
  shows «*Curry*[*a*, *b*, *c*] *g* : *a* → *y*»
    ⟨*proof*⟩

lemma *Curry-simps* [*simp*]:
assumes *ide a* **and** *ide b* **and** «$g : a \otimes b \rightarrow c$»
shows *arr* (*Curry*[*a*, *b*, *c*] *g*)
and *dom* (*Curry*[*a*, *b*, *c*] *g*) = *a*
and *cod* (*Curry*[*a*, *b*, *c*] *g*) = *exp b c*
⟨*proof*⟩

lemma *eval-in-hom*$_{ECMC}$ [*intro*]:
assumes *ide b* **and** *ide c* **and** $x = exp\ b\ c \otimes b$
shows «*eval b c* : $x \rightarrow c$»
⟨*proof*⟩

lemma *eval-simps* [*simp*]:
assumes *ide b* **and** *ide c*
shows *arr* (*eval b c*) **and** *dom* (*eval b c*) = *exp b c* ⊗ *b* **and** *cod* (*eval b c*) = *c*
⟨*proof*⟩

lemma *Uncurry-in-hom* [*intro*]:
assumes *ide b* **and** *ide c* **and** «$f : a \rightarrow exp\ b\ c$» **and** $x = a \otimes b$
shows «*Uncurry*[*b*, *c*] *f* : $x \rightarrow c$»
⟨*proof*⟩

lemma *Uncurry-simps* [*simp*]:
assumes *ide b* **and** *ide c* **and** «$f : a \rightarrow exp\ b\ c$»
shows *arr* (*Uncurry*[*b*, *c*] *f*)
and *dom* (*Uncurry*[*b*, *c*] *f*) = *a* ⊗ *b*
and *cod* (*Uncurry*[*b*, *c*] *f*) = *c*
⟨*proof*⟩

lemma *Uncurry-exp*:
assumes *ide a* **and** *ide b*
shows *Uncurry*[*a*, *b*] (*exp a b*) = *eval a b*
⟨*proof*⟩

lemma *comp-Curry-arr*:
assumes *ide b* **and** «$f : x \rightarrow a$» **and** «$g : a \otimes b \rightarrow c$»
shows *Curry*[*a*, *b*, *c*] *g* · *f* = *Curry*[*x*, *b*, *c*] (*g* · (*f* ⊗ *b*))
⟨*proof*⟩

lemma *terminal-arrow-from-functor-eval*:
assumes *ide b* **and** *ide c*
shows *terminal-arrow-from-functor C C* ($\lambda x.$ *T* (*x*, *b*)) (*exp b c*) *c* (*eval b c*)
⟨*proof*⟩

lemma *is-closed-monoidal-category*:
shows *closed-monoidal-category C T* α ι
⟨*proof*⟩

lemma *retraction-eval-ide-self*:

assumes *ide a*
shows *retraction* (*eval a a*)
⟨*proof*⟩

end

context *elementary-closed-symmetric-monoidal-category*
begin

lemma *is-closed-symmetric-monoidal-category*:
shows *closed-symmetric-monoidal-category C T* α ι σ
⟨*proof*⟩

end

1.1.2 A CMC Extends to an ECMC

context *closed-monoidal-category*
begin

lemma *has-exponentials*:
assumes *ide b* **and** *ide c*
shows $\exists\, x\ e.\ ide\ x \wedge «e : x \otimes b \rightarrow c» \wedge$
$(\forall\, a\ g.\ ide\ a \wedge$
$«g : a \otimes b \rightarrow c» \longrightarrow (\exists!f.\ «f : a \rightarrow x» \wedge g = e \cdot (f \otimes b)))$
⟨*proof*⟩

definition *some-exp* (‹$exp^?$›)
where $exp^?\ b\ c \equiv SOME\ x.\ ide\ x \wedge$
$(\exists\, e.\ «e : x \otimes b \rightarrow c» \wedge$
$(\forall\, a\ g.\ ide\ a \wedge «g : a \otimes b \rightarrow c»$
$\longrightarrow (\exists!f.\ «f : a \rightarrow x» \wedge g = e \cdot (f \otimes b))))$

definition *some-eval* (‹$eval^?$›)
where $eval^?\ b\ c \equiv SOME\ e.\ «e : exp^?\ b\ c \otimes b \rightarrow c» \wedge$
$(\forall\, a\ g.\ ide\ a \wedge «g : a \otimes b \rightarrow c»$
$\longrightarrow (\exists!f.\ «f : a \rightarrow exp^?\ b\ c» \wedge g = e \cdot (f \otimes b)))$

definition *some-Curry* (‹$Curry^?$[-, -, -]›)
where $Curry^?[a, b, c]\ g \equiv$
$THE\ f.\ «f : a \rightarrow exp^?\ b\ c» \wedge g = eval^?\ b\ c \cdot (f \otimes b)$

abbreviation *some-Uncurry* (‹$Uncurry^?$[-, -]›)
where $Uncurry^?[b, c]\ f \equiv eval^?\ b\ c \cdot (f \otimes b)$

lemma *Curry-uniqueness*:
assumes *ide b* **and** *ide c*

shows *ide* ($exp^?$ *b c*) **and** «$eval^?$ *b c* : $exp^?$ *b c* ⊗ *b* → *c*»
and ⟦ *ide a*; «*g* : *a* ⊗ *b* → *c*» ⟧
⟹ ∃!*f*. «*f* : *a* → $exp^?$ *b c*» ∧ *g* = $Uncurry^?$[*b*, *c*] *f*
⟨*proof*⟩

lemma *some-eval-in-hom* [*intro*]:
assumes *ide b* **and** *ide c* **and** *x* = $exp^?$ *b c* ⊗ *b*
shows «$eval^?$ *b c* : *x* → *c*»
⟨*proof*⟩

lemma *some-Uncurry-some-Curry*:
assumes *ide a* **and** *ide b* **and** «*g* : *a* ⊗ *b* → *c*»
shows «$Curry^?$[*a*, *b*, *c*] *g* : *a* → $exp^?$ *b c*»
and $Uncurry^?$[*b*, *c*] ($Curry^?$[*a*, *b*, *c*] *g*) = *g*
⟨*proof*⟩

lemma *some-Curry-some-Uncurry*:
assumes *ide b* **and** *ide c* **and** «*h* : *a* → $exp^?$ *b c*»
shows $Curry^?$[*a*, *b*, *c*] ($Uncurry^?$[*b*, *c*] *h*) = *h*
⟨*proof*⟩

lemma $extends\text{-}to\text{-}elementary\text{-}closed\text{-}monoidal\text{-}category_{CMC}$:
shows *elementary-closed-monoidal-category*
C T α ι some-exp some-eval some-Curry
⟨*proof*⟩

end

context *closed-symmetric-monoidal-category*
begin

lemma $extends\text{-}to\text{-}elementary\text{-}closed\text{-}symmetric\text{-}monoidal\text{-}category_{CMC}$:
shows *elementary-closed-symmetric-monoidal-category*
C T α ι σ some-exp some-eval some-Curry
⟨*proof*⟩

end

1.2 Internal Hom Functors

For each object *x* of a closed monoidal category *C*, we can define a covariant endofunctor $Exp^{\rightarrow}$ *x* − of *C*, which takes each arrow *g* to an arrow «$Exp^{\rightarrow}$ *x g* : *exp x* (*dom g*) → *exp x* (*cod g*)». Similarly, for each object *y*, we can define a contravariant endofunctor $Exp^{\leftarrow}$ − *y* of *C*, which takes each arrow *f* of C^{op} to an arrow «$Exp^{\leftarrow}$ *f y* : *exp* (*cod f*) *y* → *exp* (*dom f*) *y*» of *C*. These two endofunctors commute with each other and compose to form a single binary "internal hom" functor *Exp* from $C^{op} \times C$ to *C*.

context *elementary-closed-monoidal-category*
begin

abbreviation *cov-Exp* (‹$Exp^{\rightarrow}$›)
where $Exp^{\rightarrow}$ x g ≡ *if arr g*
then Curry[*exp x* (*dom g*), *x*, *cod g*] (*g* · *eval x* (*dom g*))
else null

abbreviation *cnt-Exp* (‹$Exp^{\leftarrow}$›)
where $Exp^{\leftarrow}$ f y ≡ *if arr f*
then Curry[*exp* (*cod f*) *y*, *dom f*, *y*]
(*eval* (*cod f*) *y* · (*exp* (*cod f*) *y* ⊗ *f*))
else null

lemma *cov-Exp-in-hom*:
assumes *ide x* **and** *arr g*
shows «$Exp^{\rightarrow}$ *x g* : *exp x* (*dom g*) → *exp x* (*cod g*)»
⟨*proof*⟩

lemma *cnt-Exp-in-hom*:
assumes *arr f* **and** *ide y*
shows «$Exp^{\leftarrow}$ *f y* : *exp* (*cod f*) *y* → *exp* (*dom f*) *y*»
⟨*proof*⟩

lemma *cov-Exp-ide*:
assumes *ide a* **and** *ide b*
shows $Exp^{\rightarrow}$ *a b* = *exp a b*
⟨*proof*⟩

lemma *cnt-Exp-ide*:
assumes *ide a* **and** *ide b*
shows $Exp^{\leftarrow}$ *a b* = *exp a b*
⟨*proof*⟩

lemma *cov-Exp-comp*:
assumes *ide x* **and** *seq g f*
shows $Exp^{\rightarrow}$ *x* (*g* · *f*) = $Exp^{\rightarrow}$ *x g* · $Exp^{\rightarrow}$ *x f*
⟨*proof*⟩

lemma *cnt-Exp-comp*:
assumes *seq g f* **and** *ide y*
shows $Exp^{\leftarrow}$ (*g* · *f*) *y* = $Exp^{\leftarrow}$ *f y* · $Exp^{\leftarrow}$ *g y*
⟨*proof*⟩

lemma *functor-cov-Exp*:
assumes *ide x*
shows *functor C C* ($Exp^{\rightarrow}$ *x*)
⟨*proof*⟩

interpretation *Cop*: *dual-category C* ⟨*proof*⟩

lemma *functor-cnt-Exp*:
assumes *ide x*
shows *functor Cop.comp C* ($\lambda f.$ *Exp*$^{\leftarrow}$ *f x*)
⟨*proof*⟩

lemma *cov-cnt-Exp-commute*:
assumes *arr f* **and** *arr g*
shows *Exp*$^{\rightarrow}$ (*dom f*) *g* · *Exp*$^{\leftarrow}$ *f* (*dom g*) =
Exp$^{\leftarrow}$ *f* (*cod g*) · *Exp*$^{\rightarrow}$ (*cod f*) *g*
⟨*proof*⟩

definition *Exp*
where *Exp f g* ≡ *Exp*$^{\rightarrow}$ (*dom f*) *g* · *Exp*$^{\leftarrow}$ *f* (*dom g*)

lemma *Exp-in-hom*:
assumes *arr f* **and** *arr g*
shows «*Exp f g* : *Exp* (*cod f*) (*dom g*) → *Exp* (*dom f*) (*cod g*)»
⟨*proof*⟩

lemma *Exp-ide*:
assumes *ide a* **and** *ide b*
shows *Exp a b* = *exp a b*
⟨*proof*⟩

lemma *Exp-comp*:
assumes *seq g f* **and** *seq k h*
shows *Exp* (*g* · *f*) (*k* · *h*) = *Exp f k* · *Exp g h*
⟨*proof*⟩

interpretation *CopxC*: *product-category Cop.comp C* ⟨*proof*⟩

lemma *functor-Exp*:
shows *binary-functor Cop.comp C C* ($\lambda fg.$ *Exp* (*fst fg*) (*snd fg*))
⟨*proof*⟩

lemma *Exp-x-ide*:
assumes *ide y*
shows ($\lambda x.$ *Exp x y*) = ($\lambda x.$ *Exp*$^{\leftarrow}$ *x y*)
⟨*proof*⟩

lemma *Exp-ide-y*:
assumes *ide x*
shows ($\lambda y.$ *Exp x y*) = ($\lambda y.$ *Exp*$^{\rightarrow}$ *x y*)
⟨*proof*⟩

lemma *Uncurry-Exp-dom*:
assumes *arr f*

shows *Uncurry* (*dom f*) (*cod f*) (*Exp* (*dom f*) *f*) = *f* · *eval* (*dom f*) (*dom f*)
⟨*proof*⟩

1.2.1 Exponentiation by Unity

In this section we define and develop the properties of inverse arrows *Up a* : *a* → *exp* $\mathcal{I}$ *a* and *Dn a* : *exp* $\mathcal{I}$ *a* → *a*, which exist in any closed monoidal category.

interpretation *elementary-monoidal-category C tensor unity lunit runit assoc*
⟨*proof*⟩

abbreviation *Up*
where *Up a* ≡ *Curry*[*a*, $\mathcal{I}$, *a*] r[*a*]

abbreviation *Dn*
where *Dn a* ≡ *eval* $\mathcal{I}$ *a* · r^{-1}[*exp* $\mathcal{I}$ *a*]

lemma *isomorphic-exp-unity*:
assumes *ide a*
shows «*Up a* : *a* → *exp* $\mathcal{I}$ *a*»
and «*Dn a* : *exp* $\mathcal{I}$ *a* → *a*»
and *inverse-arrows* (*Up a*) (*Dn a*)
and *isomorphic* (*exp* $\mathcal{I}$ *a*) *a*
⟨*proof*⟩

The maps *Up* and *Dn* are natural in a.

lemma *Up-Dn-naturality*:
assumes *arr f*
shows $Exp^{\rightarrow}$ $\mathcal{I}$ *f* · *Up* (*dom f*) = *Up* (*cod f*) · *f*
and *Dn* (*cod f*) · $Exp^{\rightarrow}$ $\mathcal{I}$ *f* = *f* · *Dn* (*dom f*)
⟨*proof*⟩

1.2.2 Internal Currying

Currying internalizes to an isomorphism between *exp* (*x* ⊗ *a*) *b* and *exp x* (*exp a b*).

abbreviation *curry*
where *curry x b c* ≡
  Curry[*exp* (*x* ⊗ *b*) *c*, *x*, *exp b c*]
    (*Curry*[*exp* (*x* ⊗ *b*) *c* ⊗ *x*, *b*, *c*]
      (*eval* (*x* ⊗ *b*) *c* · a[*exp* (*x* ⊗ *b*) *c*, *x*, *b*]))

abbreviation *uncurry*
where *uncurry x b c* ≡
  Curry[*exp x* (*exp b c*), *x* ⊗ *b*, *c*]
    (*eval b c* · (*eval x* (*exp b c*) ⊗ *b*) · a^{-1}[*exp x* (*exp b c*), *x*, *b*])

lemma *internal-curry*:

assumes *ide x* **and** *ide a* **and** *ide b*
shows «*curry x a b* : *exp* (*x* ⊗ *a*) *b* → *exp x* (*exp a b*)»
and «*uncurry x a b* : *exp x* (*exp a b*) → *exp* (*x* ⊗ *a*) *b*»
and *inverse-arrows* (*curry x a b*) (*uncurry x a b*)
⟨*proof*⟩

Internal currying and uncurrying are the components of natural isomorphisms between the contravariant functors $Exp^{\leftarrow}$ (- ⊗ *b*) *c* and $Exp^{\leftarrow}$ - (*exp b c*).

lemma *uncurry-naturality*:
assumes *ide b* **and** *ide c* **and** *Cop.arr f*
shows *uncurry* (*Cop.cod f*) *b c* · $Exp^{\leftarrow}$ *f* (*exp b c*) =
Curry[*exp* (*Cop.dom f*) (*exp b c*), *Cop.cod f* ⊗ *b*, *c*]
(*eval* (*Cop.dom f* ⊗ *b*) *c* · (*uncurry* (*Cop.dom f*) *b c* ⊗ *f* ⊗ *b*))
and $Exp^{\leftarrow}$ (*f* ⊗ *b*) *c* · *uncurry* (*Cop.dom f*) *b c* =
Curry[*exp* (*Cop.dom f*) (*exp b c*), *Cop.cod f* ⊗ *b*, *c*]
(*eval* (*Cop.dom f* ⊗ *b*) *c* · (*uncurry* (*Cop.dom f*) *b c* ⊗ *f* ⊗ *b*))
and *uncurry* (*Cop.cod f*) *b c* · $Exp^{\leftarrow}$ *f* (*exp b c*) =
$Exp^{\leftarrow}$ (*f* ⊗ *b*) *c* · *uncurry* (*Cop.dom f*) *b c*
⟨*proof*⟩

lemma *natural-isomorphism-uncurry*:
assumes *ide b* **and** *ide c*
shows *natural-isomorphism Cop.comp C*
(λ*x*. $Exp^{\leftarrow}$ *x* (*exp b c*)) (λ*x*. $Exp^{\leftarrow}$ (*x* ⊗ *b*) *c*)
(λ*f*. *uncurry* (*Cop.cod f*) *b c* · $Exp^{\leftarrow}$ *f* (*exp b c*))
⟨*proof*⟩

lemma *natural-isomorphism-curry*:
assumes *ide b* **and** *ide c*
shows *natural-isomorphism Cop.comp C*
(λ*x*. $Exp^{\leftarrow}$ (*x* ⊗ *b*) *c*) (λ*x*. $Exp^{\leftarrow}$ *x* (*exp b c*))
(λ*f*. *curry* (*Cop.cod f*) *b c* · $Exp^{\leftarrow}$ (*f* ⊗ *b*) *c*)
⟨*proof*⟩

1.2.3 Yoneda Embedding

The internal hom provides a closed monoidal category *C* with a "Yoneda embedding", which is a mapping that takes each arrow *g* of *C* to a natural transformation from the contravariant functor $Exp^{\leftarrow}$ - (*dom g*) to the contravariant functor $Exp^{\leftarrow}$ - (*cod g*). Note that here the target category is *C* itself, not a category of sets and functions as in the classical case. Note also that we are talking here about ordinary functors and natural transformations. We can easily prove from general considerations that the Yoneda embedding (so-defined) is faithful. However, to obtain a fullness result requires the development of a certain amount of enriched category theory, which we do elsewhere.

lemma *yoneda-embedding*:

assumes «$g : a \rightarrow b$»
shows *natural-transformation Cop.comp C*
$(\lambda x.\ Exp^{\leftarrow}\ x\ a)\ (\lambda x.\ Exp^{\leftarrow}\ x\ b)\ (\lambda x.\ Exp\ x\ g)$
and $Uncurry[a,\ b]\ (Exp\ a\ g \cdot Curry[\mathcal{I},\ a,\ a]\ \mathrm{l}[a]) \cdot \mathrm{l}^{-1}[a] = g$
⟨*proof*⟩

lemma *yoneda-embedding-is-faithful*:
assumes *par g g′* **and** $(\lambda x.\ Exp\ x\ g) = (\lambda x.\ Exp\ x\ g')$
shows $g = g'$
⟨*proof*⟩

The following is a version of the key fact underlying the classical Yoneda Lemma: for any natural transformation τ from $Exp^{\leftarrow}$ - a to $Exp^{\leftarrow}$ - b, there is a fixed arrow $g : a \rightarrow b$, depending only on the single component $\tau\ a$, such that the compositions $\tau\ x \cdot e$ of an arbitrary component $\tau\ x$ with arbitrary global elements $e : \mathcal{I} \rightarrow exp\ x\ a$ depend on τ only via g, and hence only via $\tau\ a$.

lemma *hom-transformation-expansion*:
assumes *natural-transformation*
Cop.comp C $(\lambda x.\ Exp^{\leftarrow}\ x\ a)\ (\lambda x.\ Exp^{\leftarrow}\ x\ b)\ \tau$
and *ide a* **and** *ide b*
shows «$Uncurry[a,\ b]\ (\tau\ a \cdot Curry[\mathcal{I},\ a,\ a]\ \mathrm{l}[a]) \cdot \mathrm{l}^{-1}[a] : a \rightarrow b$»
and $\bigwedge x\ e.\ [\![ide\ x;$ «$e : \mathcal{I} \rightarrow exp\ x\ a$»$]\!] \Longrightarrow$
$\tau\ x \cdot e = Exp\ x\ (Uncurry[a,\ b]\ (\tau\ a \cdot Curry[\mathcal{I},\ a,\ a]\ \mathrm{l}[a]) \cdot \mathrm{l}^{-1}[a]) \cdot e$
⟨*proof*⟩

1.3 Enriched Structure

In this section we do the main work involved in showing that a closed monoidal category is "enriched in itself". For this, we need to define, for each object a, an arrow $Id\ a : \mathcal{I} \rightarrow exp\ a\ a$ to serve as the "identity at a", and for every three objects a, b, and c, a "compositor" $Comp\ a\ b\ c : exp\ b\ c \otimes exp\ a\ b \rightarrow exp\ a\ c$. We also need to prove that these satisfy the appropriate unit and associativity laws. Although essentially all the work is done here, the statement and proof of the the final result is deferred to a separate theory *EnrichedCategory* so that a mutual dependence between that theory and the present one is not introduced.

interpretation *elementary-monoidal-category C tensor unity lunit runit assoc*
⟨*proof*⟩

definition *Id*
where $Id\ a \equiv Curry[\mathcal{I},\ a,\ a]\ \mathrm{l}[a]$

lemma *Id-in-hom* [*intro*]:
assumes *ide a*
shows «$Id\ a : \mathcal{I} \rightarrow exp\ a\ a$»
⟨*proof*⟩

lemma *Id-simps* [*simp*]:
assumes *ide a*
shows *arr* (*Id a*)
and *dom* (*Id a*) = $\mathcal{I}$
and *cod* (*Id a*) = *exp a a*
⟨*proof*⟩

The next definition follows Kelly [1], section 1.6.

definition *Comp*
where *Comp a b c* ≡
Curry[*exp b c* ⊗ *exp a b*, *a*, *c*]
(*eval b c* · (*exp b c* ⊗ *eval a b*) · a[*exp b c*, *exp a b*, *a*])

lemma *Comp-in-hom* [*intro*]:
assumes *ide a* **and** *ide b* **and** *ide c*
shows «*Comp a b c* : *exp b c* ⊗ *exp a b* → *exp a c*»
⟨*proof*⟩

lemma *Comp-simps* [*simp*]:
assumes *ide a* **and** *ide b* **and** *ide c*
shows *arr* (*Comp a b c*)
and *dom* (*Comp a b c*) = *exp b c* ⊗ *exp a b*
and *cod* (*Comp a b c*) = *exp a c*
⟨*proof*⟩

lemma *Comp-unit-right*:
assumes *ide a* **and** *ide b* **and** *ide c*
shows «*Comp a a b* · (*exp a b* ⊗ *Id a*) : *exp a b* ⊗ $\mathcal{I}$ → *exp a b*»
and *Comp a a b* · (*exp a b* ⊗ *Id a*) = r[*exp a b*]
⟨*proof*⟩

lemma *Comp-unit-left*:
assumes *ide a* **and** *ide b* **and** *ide c*
shows «*Comp a b b* · (*Id b* ⊗ *exp a b*) : $\mathcal{I}$ ⊗ *exp a b* → *exp a b*»
and *Comp a b b* · (*Id b* ⊗ *exp a b*) = l[*exp a b*]
⟨*proof*⟩

lemma *Comp-assoc*$_{ECMC}$:
assumes *ide a* **and** *ide b* **and** *ide c* **and** *ide d*
shows «*Comp a b d* · (*Comp b c d* ⊗ *exp a b*) :
(*exp c d* ⊗ *exp b c*) ⊗ *exp a b* → *exp a d*»
and *Comp a b d* · (*Comp b c d* ⊗ *exp a b*) =
Comp a c d · (*exp c d* ⊗ *Comp a b c*) · a[*exp c d*, *exp b c*, *exp a b*]
⟨*proof*⟩

end

end

1.4 Cartesian Closed Monoidal Categories

A *cartesian closed monoidal category* is a cartesian monoidal category that is a closed monoidal category with respect to a chosen product. This is not quite the same thing as a cartesian closed category, because a cartesian monoidal category (being a monoidal category) has chosen structure (the tensor, associators, and unitors), whereas we have defined a cartesian closed category to be an abstract category satisfying certain properties that are expressed without assuming any chosen structure.

theory *CartesianClosedMonoidalCategory*
imports *Category3.CartesianClosedCategory MonoidalCategory.CartesianMonoidalCategory ClosedMonoidalCategory*
begin

locale *cartesian-closed-monoidal-category* =
cartesian-monoidal-category +
closed-monoidal-category

locale *elementary-cartesian-closed-monoidal-category* =
cartesian-monoidal-category +
elementary-closed-monoidal-category
begin

lemmas *prod-eq-tensor* [*simp*]

end

The following is the main purpose for the current theory: to show that a cartesian closed category with chosen structure determines a cartesian closed monoidal category.

context *elementary-cartesian-closed-category*
begin

interpretation *CMC*: *cartesian-monoidal-category C Prod* α ι
⟨*proof*⟩

interpretation *CMC*: *closed-monoidal-category C Prod* α ι
⟨*proof*⟩

lemma *extends-to-closed-monoidal-category*$_{ECCC}$:
shows *closed-monoidal-category C Prod* α ι
⟨*proof*⟩

lemma *extends-to-cartesian-closed-monoidal-category*$_{ECCC}$:
shows *cartesian-closed-monoidal-category C Prod* α ι
⟨*proof*⟩

interpretation *CMC*: *elementary-monoidal-category*

C CMC.tensor CMC.unity CMC.lunit CMC.runit CMC.assoc
⟨*proof*⟩

interpretation *CMC*: *elementary-closed-monoidal-category*
C Prod α ι *exp eval curry*
⟨*proof*⟩

lemma *extends-to-elementary-closed-monoidal-category*$_{ECCC}$:
shows *elementary-closed-monoidal-category C Prod* α ι *exp eval curry*
⟨*proof*⟩

lemma *extends-to-elementary-cartesian-closed-monoidal-category*$_{ECCC}$:
shows *elementary-cartesian-closed-monoidal-category C Prod* α ι *exp eval curry*
⟨*proof*⟩

end

context *elementary-cartesian-closed-monoidal-category*
begin

interpretation *elementary-monoidal-category C tensor unity lunit runit assoc*
⟨*proof*⟩

The following fact is not used in the present article, but it is a natural and likely useful lemma for which I constructed a proof at one point. The proof requires cartesianness; I suspect this is essential, but I am not absolutely certain of it.

lemma *isomorphic-exp-prod*:
assumes *ide a* **and** *ide b* **and** *ide x*
shows «$\langle Curry[exp\ x\ (a \otimes b),\ x,\ a]\ (\mathfrak{p}_1[a,\ b] \cdot eval\ x\ (a \otimes b)),$
$Curry[exp\ x\ (a \otimes b),\ x,\ b]\ (\mathfrak{p}_0[a,\ b] \cdot eval\ x\ (a \otimes b))\rangle$
$:\ exp\ x\ (a \otimes b) \rightarrow exp\ x\ a \otimes exp\ x\ b$»
(**is** «$\langle ?C,\ ?D\rangle : exp\ x\ (a \otimes b) \rightarrow exp\ x\ a \otimes exp\ x\ b$»)
and «$Curry[exp\ x\ a \otimes exp\ x\ b,\ x,\ a \otimes b]$
$\langle eval\ x\ a \cdot \langle \mathfrak{p}_1[exp\ x\ a,\ exp\ x\ b] \cdot \mathfrak{p}_1[exp\ x\ a \otimes exp\ x\ b,\ x],$
$\mathfrak{p}_0[exp\ x\ a \otimes exp\ x\ b,\ x]\rangle,$
$eval\ x\ b \cdot \langle \mathfrak{p}_0[exp\ x\ a,\ exp\ x\ b] \cdot \mathfrak{p}_1[exp\ x\ a \otimes exp\ x\ b,\ x],$
$\mathfrak{p}_0[exp\ x\ a \otimes exp\ x\ b,\ x]\rangle\rangle$
$:\ exp\ x\ a \otimes exp\ x\ b \rightarrow exp\ x\ (a \otimes b)$»
(**is** «$Curry[exp\ x\ a \otimes exp\ x\ b,\ x,\ a \otimes b]\ \langle ?A,\ ?B\rangle$
$:\ exp\ x\ a \otimes exp\ x\ b \rightarrow exp\ x\ (a \otimes b)$»)
and *inverse-arrows*
$(Curry[exp\ x\ a \otimes exp\ x\ b,\ x,\ a \otimes b]$
$\langle eval\ x\ a \cdot \langle \mathfrak{p}_1[exp\ x\ a,\ exp\ x\ b] \cdot \mathfrak{p}_1[exp\ x\ a \otimes exp\ x\ b,\ x],$
$\mathfrak{p}_0[exp\ x\ a \otimes exp\ x\ b,\ x]\rangle,$
$eval\ x\ b \cdot \langle \mathfrak{p}_0[exp\ x\ a,\ exp\ x\ b] \cdot \mathfrak{p}_1[exp\ x\ a \otimes exp\ x\ b,\ x],$
$\mathfrak{p}_0[exp\ x\ a \otimes exp\ x\ b,\ x]\rangle\rangle)$
$\langle Curry[exp\ x\ (a \otimes b),\ x,\ a]\ (\mathfrak{p}_1[a,\ b] \cdot eval\ x\ (a \otimes b)),$
$Curry[exp\ x\ (a \otimes b),\ x,\ b]\ (\mathfrak{p}_0[a,\ b] \cdot eval\ x\ (a \otimes b))\rangle$

and *isomorphic* (*exp x* (*a* ⊗ *b*)) (*exp x a* ⊗ *exp x b*)
⟨*proof*⟩

end

end

Chapter 2

Enriched Categories

The notion of an enriched category [1] generalizes the concept of category by replacing the hom-sets of an ordinary category by objects of an arbitrary monoidal category M. The choice, for each object a, of a distinguished element $id\ a : a \rightarrow a$ as an identity, is replaced by an arrow $Id\ a : \mathcal{I} \rightarrow Hom\ a\ a$ of M. The composition operation is similarly replaced by a family of arrows $Comp\ a\ b\ c : Hom\ B\ C \otimes Hom\ A\ B \rightarrow Hom\ A\ C$ of M. The identity and composition are required to satisfy unit and associativity laws which are expressed as commutative diagrams in M.

theory *EnrichedCategory*
imports *ClosedMonoidalCategory*
begin

context *monoidal-category*
begin

abbreviation ι' $(\langle\iota^{-1}\rangle)$
where $\iota' \equiv inv\ \iota$

end

context *elementary-symmetric-monoidal-category*
begin

lemma *sym-unit*:
shows $\iota \cdot \mathrm{s}[\mathcal{I}, \mathcal{I}] = \iota$
⟨*proof*⟩

lemma *sym-inv-unit*:
shows $\mathrm{s}[\mathcal{I}, \mathcal{I}] \cdot inv\ \iota = inv\ \iota$
⟨*proof*⟩

end

2.1 Basic Definitions

```
locale enriched-category =
  monoidal-category +
fixes Obj :: 'o set
and Hom :: 'o ⇒ 'o ⇒ 'a
and Id :: 'o ⇒ 'a
and Comp :: 'o ⇒ 'o ⇒ 'o ⇒ 'a
assumes ide-Hom [intro, simp]: ⟦a ∈ Obj; b ∈ Obj⟧ ⟹ ide (Hom a b)
  and Id-in-hom [intro]: a ∈ Obj ⟹ «Id a : ℐ → Hom a a»
  and Comp-in-hom [intro]: ⟦a ∈ Obj; b ∈ Obj; c ∈ Obj⟧ ⟹
                           «Comp a b c : Hom b c ⊗ Hom a b → Hom a c»
  and Comp-Hom-Id: ⟦a ∈ Obj; b ∈ Obj⟧ ⟹
                    Comp a a b · (Hom a b ⊗ Id a) = r[Hom a b]
  and Comp-Id-Hom: ⟦a ∈ Obj; b ∈ Obj⟧ ⟹
                    Comp a b b · (Id b ⊗ Hom a b) = l[Hom a b]
  and Comp-assoc: ⟦a ∈ Obj; b ∈ Obj; c ∈ Obj; d ∈ Obj⟧ ⟹
                   Comp a b d · (Comp b c d ⊗ Hom a b) =
                   Comp a c d · (Hom c d ⊗ Comp a b c) ·
                    a[Hom c d, Hom b c, Hom a b]
```

A functor from an enriched category A to an enriched category B consists of an object map $F_o : Obj_A \rightarrow Obj_B$ and a map F_a that assigns to each pair of objects a b in Obj_A an arrow $F_a\ a\ b : Hom_A\ a\ b \rightarrow Hom_B\ (F_o\ a)\ (F_o\ b)$ of the underlying monoidal category, subject to equations expressing that identities and composition are preserved.

```
locale enriched-functor =
  monoidal-category C T α ι +
  A: enriched-category C T α ι Obj_A Hom_A Id_A Comp_A +
  B: enriched-category C T α ι Obj_B Hom_B Id_B Comp_B
for C :: 'm ⇒ 'm ⇒ 'm  (infixr ‹·› 55)
and T :: 'm × 'm ⇒ 'm
and α :: 'm × 'm × 'm ⇒ 'm
and ι :: 'm
and Obj_A :: 'a set
and Hom_A :: 'a ⇒ 'a ⇒ 'm
and Id_A :: 'a ⇒ 'm
and Comp_A :: 'a ⇒ 'a ⇒ 'a ⇒ 'm
and Obj_B :: 'b set
and Hom_B :: 'b ⇒ 'b ⇒ 'm
and Id_B :: 'b ⇒ 'm
and Comp_B :: 'b ⇒ 'b ⇒ 'b ⇒ 'm
and F_o :: 'a ⇒ 'b
and F_a :: 'a ⇒ 'a ⇒ 'm +
assumes extensionality: a ∉ Obj_A ∨ b ∉ Obj_A ⟹ F_a a b = null
assumes preserves-Obj [intro]: a ∈ Obj_A ⟹ F_o a ∈ Obj_B
```

and *preserves-Hom*: $[\![a \in Obj_A;\ b \in Obj_A]\!] \Longrightarrow$
«$F_a\ a\ b : Hom_A\ a\ b \rightarrow Hom_B\ (F_o\ a)\ (F_o\ b)$»
and *preserves-Id*: $a \in Obj_A \Longrightarrow F_a\ a\ a \cdot Id_A\ a = Id_B\ (F_o\ a)$
and *preserves-Comp*: $[\![a \in Obj_A;\ b \in Obj_A;\ c \in Obj_A]\!] \Longrightarrow$
$Comp_B\ (F_o\ a)\ (F_o\ b)\ (F_o\ c) \cdot T\ (F_a\ b\ c,\ F_a\ a\ b) =$
$F_a\ a\ c \cdot Comp_A\ a\ b\ c$

locale *fully-faithful-enriched-functor* =
enriched-functor +
assumes *locally-iso*: $[\![a \in Obj_A;\ b \in Obj_A]\!] \Longrightarrow iso\ (F_a\ a\ b)$

A natural transformation from an an enriched functor $F = (F_o,\ F_a)$ to an enriched functor $G = (G_o,\ G_a)$ consists of a map τ that assigns to each object $a \in Obj_A$ a "component at a", which is an arrow $\tau\ a : \mathcal{I} \rightarrow Hom_B\ (F_o\ a)\ (G_o\ a)$, subject to an equation that expresses the naturality condition.

locale *enriched-natural-transformation* =
monoidal-category $C\ T\ \alpha\ \iota$ +
A: *enriched-category* $C\ T\ \alpha\ \iota\ Obj_A\ Hom_A\ Id_A\ Comp_A$ +
B: *enriched-category* $C\ T\ \alpha\ \iota\ Obj_B\ Hom_B\ Id_B\ Comp_B$ +
F: *enriched-functor* $C\ T\ \alpha\ \iota$
$Obj_A\ Hom_A\ Id_A\ Comp_A\ Obj_B\ Hom_B\ Id_B\ Comp_B\ F_o\ F_a$ +
G: *enriched-functor* $C\ T\ \alpha\ \iota$
$Obj_A\ Hom_A\ Id_A\ Comp_A\ Obj_B\ Hom_B\ Id_B\ Comp_B\ G_o\ G_a$
for $C :: 'm \Rightarrow 'm \Rightarrow 'm$ (**infixr** ‹·› *55*)
and $T :: 'm \times 'm \Rightarrow 'm$
and $\alpha :: 'm \times 'm \times 'm \Rightarrow 'm$
and $\iota :: 'm$
and $Obj_A :: 'a\ set$
and $Hom_A :: 'a \Rightarrow 'a \Rightarrow 'm$
and $Id_A :: 'a \Rightarrow 'm$
and $Comp_A :: 'a \Rightarrow 'a \Rightarrow 'a \Rightarrow 'm$
and $Obj_B :: 'b\ set$
and $Hom_B :: 'b \Rightarrow 'b \Rightarrow 'm$
and $Id_B :: 'b \Rightarrow 'm$
and $Comp_B :: 'b \Rightarrow 'b \Rightarrow 'b \Rightarrow 'm$
and $F_o :: 'a \Rightarrow 'b$
and $F_a :: 'a \Rightarrow 'a \Rightarrow 'm$
and $G_o :: 'a \Rightarrow 'b$
and $G_a :: 'a \Rightarrow 'a \Rightarrow 'm$
and $\tau :: 'a \Rightarrow 'm$ +
assumes *extensionality*: $a \notin Obj_A \Longrightarrow \tau\ a = null$
and *component-in-hom* [*intro*]: $a \in Obj_A \Longrightarrow$ «$\tau\ a : \mathcal{I} \rightarrow Hom_B\ (F_o\ a)\ (G_o\ a)$»
and *naturality*: $[\![a \in Obj_A;\ b \in Obj_A]\!] \Longrightarrow$
$Comp_B\ (F_o\ a)\ (F_o\ b)\ (G_o\ b) \cdot (\tau\ b \otimes F_a\ a\ b) \cdot \mathrm{l}^{-1}[Hom_A\ a\ b] =$
$Comp_B\ (F_o\ a)\ (G_o\ a)\ (G_o\ b) \cdot (G_a\ a\ b \otimes \tau\ a) \cdot \mathrm{r}^{-1}[Hom_A\ a\ b]$

2.1.1 Self-Enrichment

context *elementary-closed-monoidal-category*
begin

Every closed monoidal category M admits a structure of enriched category, where the exponentials in M itself serve as the "hom-objects" (*cf.* [1] Section 1.6). Essentially all the work in proving this theorem has already been done in *EnrichedCategoryBasics.ClosedMonoidalCategory*.

interpretation *closed-monoidal-category*
$\langle proof \rangle$

interpretation *EC*: *enriched-category* C T α ι ‹*Collect ide*› *exp Id Comp*
$\langle proof \rangle$

theorem *is-enriched-in-itself*:
shows *enriched-category* C T α ι (*Collect ide*) *exp Id Comp*
$\langle proof \rangle$

The following mappings define a bijection between *hom a b* and *hom* $\mathcal{I}$ (*exp a b*). These have functorial properties which are encountered repeatedly.

definition *UP* (‹-$^\uparrow$› [*100*] *100*)
where $t^\uparrow \equiv$ *if arr t then Curry*[$\mathcal{I}$, *dom t*, *cod t*] ($t \cdot$ l[*dom t*]) *else null*

definition *DN*
where *DN a b t* $\equiv$ *if arr t then Uncurry*[a, b] $t \cdot \mathrm{l}^{-1}[a]$ *else null*

abbreviation *DN′* (‹-$^\downarrow$[-, -]› [*100*] *99*)
where $t^\downarrow[a, b] \equiv$ *DN a b t*

lemma *UP-DN*:
shows [*intro*]: *arr t* $\Longrightarrow$ «$t^\uparrow : \mathcal{I} \to$ *exp* (*dom t*) (*cod t*)»
and [*intro*]: ⟦*ide a*; *ide b*; «$t : \mathcal{I} \to$ *exp a b*»⟧ $\Longrightarrow$ «$t^\downarrow[a, b]$: $a \to b$»
and [*simp*]: *arr t* $\Longrightarrow$ $(t^\uparrow)^\downarrow$[*dom t*, *cod t*] $= t$
and [*simp*]: ⟦*ide a*; *ide b*; «$t : \mathcal{I} \to$ *exp a b*»⟧ $\Longrightarrow$ $(t^\downarrow[a, b])^\uparrow = t$
$\langle proof \rangle$

lemma *UP-simps* [*simp*]:
assumes *arr t*
shows *arr* ($t^\uparrow$) **and** *dom* ($t^\uparrow$) $= \mathcal{I}$ **and** *cod* ($t^\uparrow$) $=$ *exp* (*dom t*) (*cod t*)
$\langle proof \rangle$

lemma *DN-simps* [*simp*]:
assumes *ide a* **and** *ide b* **and** *arr t* **and** *dom t* $= \mathcal{I}$ **and** *cod t* $=$ *exp a b*
shows *arr* ($t^\downarrow[a, b]$) **and** *dom* ($t^\downarrow[a, b]$) $= a$ **and** *cod* ($t^\downarrow[a, b]$) $= b$
$\langle proof \rangle$

lemma *UP-ide*:

assumes *ide a*
shows $a^{\uparrow} = Id\ a$
⟨*proof*⟩

lemma *DN-Id*:
assumes *ide a*
shows $(Id\ a)^{\downarrow}[a,\ a] = a$
⟨*proof*⟩

lemma *UP-comp*:
assumes *seq t u*
shows $(t \cdot u)^{\uparrow} = Comp\ (dom\ u)\ (cod\ u)\ (cod\ t) \cdot (t^{\uparrow} \otimes u^{\uparrow}) \cdot \iota^{-1}$
⟨*proof*⟩

end

2.2 Underlying Category, Functor, and Natural Transformation

2.2.1 Underlying Category

The underlying category (*cf.* [1] Section 1.3) of an enriched category has as its arrows from a to b the arrows $\mathcal{I} \rightarrow Hom\ a\ b$ of M (*i.e.* the points of $Hom\ a\ b$). The identity at a is $Id\ a$. The composition of arrows f and g is given by the formula: $Comp\ a\ b\ c \cdot (g \otimes f) \cdot \iota^{-1}$.

locale *underlying-category* =
M: *monoidal-category* +
A: *enriched-category*
begin

sublocale *concrete-category Obj* ‹$\lambda a\ b.\ M.hom\ \mathcal{I}\ (Hom\ a\ b)$› ‹*Id*›
‹$\lambda c\ b\ a\ g\ f.\ Comp\ a\ b\ c \cdot (g \otimes f) \cdot \iota^{-1}$›
⟨*proof*⟩

abbreviation *comp* (**infixr** ‹$\cdot_0$› *55*)
where $comp \equiv COMP$

lemma *hom-char*:
assumes $a \in Obj$ **and** $b \in Obj$
shows $hom\ (MkIde\ a)\ (MkIde\ b) = MkArr\ a\ b$ ‘ $M.hom\ \mathcal{I}\ (Hom\ a\ b)$
⟨*proof*⟩

end

2.2.2 Underlying Functor

The underlying functor of an enriched functor $F : A \longrightarrow B$ takes an arrow «$f : a \rightarrow a'$» of the underlying category A_0 (*i.e.* an arrow «$\mathcal{I} \rightarrow Hom\ a\ a'$»

of M) to the arrow «F_a a a' · f : F_o $a \to F_o$ a'» of B_0 (*i.e.* the arrow «F_a a a' · f : $\mathcal{I} \to$ *Hom* (F_o a) (F_o a')» of M).

locale *underlying-functor* =
enriched-functor
begin

sublocale A_0: *underlying-category* C T α ι Obj_A Hom_A Id_A $Comp_A$ ⟨*proof*⟩
sublocale B_0: *underlying-category* C T α ι Obj_B Hom_B Id_B $Comp_B$ ⟨*proof*⟩

notation A_0.*comp* (**infixr** ‹$\cdot_{A0}$› *55*)
notation B_0.*comp* (**infixr** ‹$\cdot_{B0}$› *55*)

definition map_0
where map_0 f = (*if* A_0.*arr* f
then B_0.*MkArr* (F_o (A_0.*Dom* f)) (F_o (A_0.*Cod* f))
(F_a (A_0.*Dom* f) (A_0.*Cod* f) · A_0.*Map* f)
else B_0.*null*)

sublocale *functor* A_0.*comp* B_0.*comp* map_0
⟨*proof*⟩

proposition *is-functor*:
shows *functor* A_0.*comp* B_0.*comp* map_0
⟨*proof*⟩

end

2.2.3 Underlying Natural Transformation

The natural transformation underlying an enriched natural transformation τ has components that are essentially those of τ, except that we have to bother ourselves about coercions between types.

locale *underlying-natural-transformation* =
enriched-natural-transformation
begin

sublocale A_0: *underlying-category* C T α ι Obj_A Hom_A Id_A $Comp_A$ ⟨*proof*⟩
sublocale B_0: *underlying-category* C T α ι Obj_B Hom_B Id_B $Comp_B$ ⟨*proof*⟩
sublocale F_0: *underlying-functor* C T α ι
Obj_A Hom_A Id_A $Comp_A$ Obj_B Hom_B Id_B $Comp_B$ F_o F_a ⟨*proof*⟩
sublocale G_0: *underlying-functor* C T α ι
Obj_A Hom_A Id_A $Comp_A$ Obj_B Hom_B Id_B $Comp_B$ G_o G_a ⟨*proof*⟩

definition map_{obj}
where map_{obj} a ≡
B_0.*MkArr* (B_0.*Dom* (F_0.map_0 a)) (B_0.*Dom* (G_0.map_0 a))
(τ (A_0.*Dom* a))

sublocale τ: *NaturalTransformation.transformation-by-components*
$A_0.comp$ $B_0.comp$ $F_0.map_0$ $G_0.map_0$ map_{obj}
⟨*proof*⟩

proposition *is-natural-transformation*:
shows *natural-transformation* $A_0.comp$ $B_0.comp$ $F_0.map_0$ $G_0.map_0$ $\tau.map$
⟨*proof*⟩

end

2.2.4 Self-Enriched Case

Here we show that a closed monoidal category C, regarded as a category enriched in itself, it is isomorphic to its own underlying category. This is useful, because it is somewhat less cumbersome to work directly in the category C than in the higher-type version that results from the underlying category construction. Kelly often regards these two categories as identical.

locale *self-enriched-category* =
elementary-closed-monoidal-category +
enriched-category C T α ι ‹*Collect ide*› *exp Id Comp*
begin

sublocale *UC*: *underlying-category* C T α ι ‹*Collect ide*› *exp Id Comp* ⟨*proof*⟩

abbreviation *toUC*
where *toUC g* ≡ *if arr g*
then UC.MkArr (*dom g*) (*cod g*) ($g^{\uparrow}$)
else UC.null

lemma *toUC-simps* [*simp*]:
assumes *arr f*
shows *UC.arr* (*toUC f*)
and *UC.dom* (*toUC f*) = *toUC* (*dom f*)
and *UC.cod* (*toUC f*) = *toUC* (*cod f*)
⟨*proof*⟩

lemma *toUC-in-hom* [*intro*]:
assumes *arr f*
shows *UC.in-hom* (*toUC f*) (*UC.MkIde* (*dom f*)) (*UC.MkIde* (*cod f*))
⟨*proof*⟩

sublocale *toUC*: *functor C UC.comp toUC*
⟨*proof*⟩

abbreviation *frmUC*
where *frmUC g* ≡ *if UC.arr g*
then ($UC.Map\ g)^{\downarrow}[UC.Dom\ g,\ UC.Cod\ g]$
else null

lemma *frmUC-simps* [*simp*]:
assumes *UC.arr f*
shows *arr* (*frmUC f*)
and *dom* (*frmUC f*) = *frmUC* (*UC.dom f*)
and *cod* (*frmUC f*) = *frmUC* (*UC.cod f*)
⟨*proof*⟩

lemma *frmUC-in-hom* [*intro*]:
assumes *UC.in-hom f a b*
shows «*frmUC f* : *frmUC a* → *frmUC b*»
⟨*proof*⟩

lemma *DN-Map-comp*:
assumes *UC.seq g f*
shows $(UC.Map\ (UC.comp\ g\ f))^{\downarrow}[UC.Dom\ f,\ UC.Cod\ g] =$
$(UC.Map\ g)^{\downarrow}[UC.Dom\ g,\ UC.Cod\ g]\ \cdot$
$(UC.Map\ f)^{\downarrow}[UC.Dom\ f,\ UC.Cod\ f]$
⟨*proof*⟩

sublocale *frmUC*: *functor UC.comp C frmUC*
⟨*proof*⟩

sublocale *inverse-functors UC.comp C toUC frmUC*
⟨*proof*⟩

lemma *inverse-functors-toUC-frmUC*:
shows *inverse-functors UC.comp C toUC frmUC*
⟨*proof*⟩

corollary *enriched-category-isomorphic-to-underlying-category*:
shows *isomorphic-categories UC.comp C*
⟨*proof*⟩

end

2.3 Opposite of an Enriched Category

Construction of the opposite of an enriched category (*cf.* [1] (1.19)) requires that the underlying monoidal category be symmetric, in order to introduce the required "twist" in the definition of composition.

locale *opposite-enriched-category* =
symmetric-monoidal-category +
EC: *enriched-category*
begin

interpretation *elementary-symmetric-monoidal-category*
C tensor unity lunit runit assoc sym

⟨*proof*⟩

abbreviation (*input*) Hom_{op}
where $Hom_{op}\ a\ b \equiv Hom\ b\ a$

abbreviation $Comp_{op}$
where $Comp_{op}\ a\ b\ c \equiv Comp\ c\ b\ a \cdot \mathrm{s}[Hom\ c\ b,\ Hom\ b\ a]$

sublocale *enriched-category C T* α ι *Obj* Hom_{op} *Id* $Comp_{op}$
⟨*proof*⟩

end

2.3.1 Relation between $(-^{op})_0$ and $(-_0)^{op}$

Kelly (comment before (1.22)) claims, for a category A enriched in a symmetric monoidal category, that we have $(A^{op})_0 = (A_0)^{op}$. This point becomes somewhat confusing, as it depends on the particular formalization one adopts for the notion of "category".

As we can see from the next two facts (*Op-UC-hom-char* and *UC-Op-hom-char*), the hom-sets *Op.UC.hom a b* and *UC.Op.hom a b* are both obtained by using *UC.MkArr* to "tag" elements of *hom* $\mathcal{I}$ (*Hom* (*UC.Dom b*) (*UC.Dom a*)) with *UC.Dom a* and *UC.Dom b*. These two hom-sets are formally distinct if (as is the case for us), the arrows of a category are regarded as containing information about their domain and codomain, so that the hom-sets are disjoint. On the other hand, if one regards a category as a collection of mappings that assign to each pair of objects a and b a corresponding set *hom a b*, then the hom-sets *Op.UC.hom a b* and *UC.Op.hom a b* could be arranged to be equal, as Kelly suggests.

locale *category-enriched-in-symmetric-monoidal-category* =
symmetric-monoidal-category +
enriched-category
begin

interpretation *elementary-symmetric-monoidal-category*
C tensor unity lunit runit assoc sym
⟨*proof*⟩

interpretation *Op*: *opposite-enriched-category C T* α ι σ *Obj Hom Id Comp*
⟨*proof*⟩
interpretation Op_0: *underlying-category C T* α ι *Obj* $Op.Hom_{op}$ *Id* $Op.Comp_{op}$
⟨*proof*⟩

interpretation *UC*: *underlying-category C T* α ι *Obj Hom Id Comp* ⟨*proof*⟩
interpretation *UC.Op*: *dual-category UC.comp* ⟨*proof*⟩

lemma *Op-UC-hom-char*:
assumes *UC.ide a* **and** *UC.ide b*
shows Op_0*.hom a b* =
UC.MkArr (*UC.Dom a*) (*UC.Dom b*) '
hom $\mathcal{I}$ (*Hom* (*UC.Dom b*) (*UC.Dom a*))
⟨*proof*⟩

lemma *UC-Op-hom-char*:
assumes *UC.ide a* **and** *UC.ide b*
shows *UC.Op.hom a b* =
UC.MkArr (*UC.Dom b*) (*UC.Dom a*) '
hom $\mathcal{I}$ (*Hom* (*UC.Dom b*) (*UC.Dom a*))
⟨*proof*⟩

abbreviation *toUCOp*
where *toUCOp f* ≡ *if* Op_0*.arr f*
then UC.MkArr (Op_0*.Cod f*) (Op_0*.Dom f*) (Op_0*.Map f*)
else UC.Op.null

sublocale *toUCOp*: *functor* Op_0*.comp UC.Op.comp toUCOp*
⟨*proof*⟩

lemma *functor-toUCOp*:
shows *functor* Op_0*.comp UC.Op.comp toUCOp*
⟨*proof*⟩

abbreviation *to*Op_0
where *to*Op_0 *f* ≡ *if UC.Op.arr f*
then Op_0*.MkArr* (*UC.Cod f*) (*UC.Dom f*) (*UC.Map f*)
else Op_0*.null*

sublocale *to*Op_0: *functor UC.Op.comp* Op_0*.comp to*Op_0
⟨*proof*⟩

lemma *functor-to*Op_0:
shows *functor UC.Op.comp* Op_0*.comp to*Op_0
⟨*proof*⟩

sublocale *inverse-functors UC.Op.comp* Op_0*.comp toUCOp to*Op_0
⟨*proof*⟩

lemma *inverse-functors-toUCOp-to*Op_0:
shows *inverse-functors UC.Op.comp* Op_0*.comp toUCOp to*Op_0
⟨*proof*⟩

end

2.4 Enriched Hom Functors

Here we exhibit covariant and contravariant hom functors as enriched functors, as in [1] Section 1.6. We don't bother to exhibit them as partial functors of a single two-argument functor, as to do so would require us to define the tensor product of enriched categories; something that would require more technology for proving coherence conditions than we have developed at present.

2.4.1 Covariant Case

locale *covariant-Hom* =
monoidal-category +

C: *elementary-closed-monoidal-category* +
enriched-category +
fixes x :: $'o$
assumes x: $x \in Obj$
begin

interpretation C: *enriched-category C T* α ι ‹*Collect ide*› *exp C.Id C.Comp*
⟨*proof*⟩
interpretation C: *self-enriched-category C T* α ι *exp eval Curry* ⟨*proof*⟩

abbreviation hom_o
where $hom_o \equiv Hom\ x$

abbreviation hom_a
where $hom_a \equiv \lambda b\ c.$ *if* $b \in Obj \wedge c \in Obj$
then $Curry[Hom\ b\ c,\ Hom\ x\ b,\ Hom\ x\ c]\ (Comp\ x\ b\ c)$
else null

sublocale *enriched-functor C T* α ι
Obj Hom Id Comp
‹*Collect ide*› *exp C.Id C.Comp*
hom_o hom_a
⟨*proof*⟩

lemma *is-enriched-functor*:
shows *enriched-functor C T* α ι
Obj Hom Id Comp
(*Collect ide*) *exp C.Id C.Comp*
hom_o hom_a
⟨*proof*⟩

sublocale C_0: *underlying-category C T* α ι ‹*Collect ide*› *exp C.Id C.Comp*
⟨*proof*⟩
sublocale *UC*: *underlying-category C T* α ι *Obj Hom Id Comp* ⟨*proof*⟩

sublocale *UF*: *underlying-functor C T* α ι
Obj Hom Id Comp
‹*Collect ide*› *exp C.Id C.Comp*
hom_o hom_a
⟨*proof*⟩

The following is Kelly's formula (1.31), for the result of applying the ordinary functor underlying the covariant hom functor, to an arrow $g : \mathcal{I} \rightarrow Hom\ b\ c$ of C_0, resulting in an arrow $Hom^{\rightarrow}\ x\ g : Hom\ x\ b \rightarrow Hom\ x\ c$ of C. The point of the result is that this can be expressed explicitly as $Comp\ x\ b\ c \cdot (g \otimes hom_o\ b) \cdot \mathrm{l}^{-1}[hom_o\ b]$. This is all very confusing at first, because Kelly identifies C with the underlying category C_0 of C regarded as a self-enriched category, whereas here we cannot ignore the fact that they are merely isomorphic via $C.frmUC$: $UC.comp \rightarrow C_0.comp$. There is also the bother that, for an arrow $g : \mathcal{I} \rightarrow Hom\ b\ c$ of C, the corresponding arrow of the underlying category UC has to be formally constructed using $UC.MkArr$, *i.e.* as $UC.MkArr\ b\ c\ g$.

lemma *Kelly-1-31*:
assumes $b \in Obj$ **and** $c \in Obj$ **and** «$g : \mathcal{I} \rightarrow Hom\ b\ c$»
shows $C.frmUC\ (UF.map_0\ (UC.MkArr\ b\ c\ g)) =$
$Comp\ x\ b\ c \cdot (g \otimes hom_o\ b) \cdot \mathrm{l}^{-1}[hom_o\ b]$
⟨*proof*⟩

abbreviation map_0
where $map_0\ b\ c\ g \equiv Comp\ x\ b\ c \cdot (g \otimes Hom\ x\ b) \cdot \mathrm{l}^{-1}[hom_o\ b]$

end

context *elementary-closed-monoidal-category*
begin

lemma *cov-Exp-DN*:
assumes «$g : \mathcal{I} \rightarrow exp\ a\ b$»
and *ide a* **and** *ide b* **and** *ide x*
shows $Exp^{\rightarrow}\ x\ (g\ {}^{\downarrow}[a, b]) =$
$(Curry[exp\ a\ b, exp\ x\ a, exp\ x\ b]\ (Comp\ x\ a\ b) \cdot g)\ {}^{\downarrow}[exp\ x\ a, exp\ x\ b]$
⟨*proof*⟩

end

2.4.2 Contravariant Case

locale *contravariant-Hom* =
symmetric-monoidal-category +
C: *elementary-closed-symmetric-monoidal-category* +
enriched-category +

fixes y :: $'o$
assumes y: $y \in Obj$
begin

interpretation C: *enriched-category* C T α ι ‹*Collect ide*› *exp* $C.Id$ $C.Comp$ ⟨*proof*⟩
interpretation C: *self-enriched-category* C T α ι *exp eval Curry* ⟨*proof*⟩

sublocale Op: *opposite-enriched-category* C T α ι σ *Obj Hom Id Comp* ⟨*proof*⟩

abbreviation hom_o
where $hom_o \equiv \lambda a.\ Hom\ a\ y$

abbreviation hom_a
where $hom_a \equiv \lambda b\ c.$ *if* $b \in Obj \wedge c \in Obj$
then $Curry[Hom\ c\ b,\ Hom\ b\ y,\ Hom\ c\ y]\ (Op.Comp_{op}\ y\ b\ c)$
else null

sublocale *enriched-functor* C T α ι
Obj $Op.Hom_{op}$ *Id* $Op.Comp_{op}$
‹*Collect ide*› *exp* $C.Id$ $C.Comp$
hom_o hom_a
⟨*proof*⟩

lemma *is-enriched-functor*:
shows *enriched-functor* C T α ι
Obj $Op.Hom_{op}$ *Id* $Op.Comp_{op}$
(*Collect ide*) *exp* $C.Id$ $C.Comp$
hom_o hom_a
⟨*proof*⟩

sublocale C_0: *underlying-category* C T α ι ‹*Collect ide*› *exp* $C.Id$ $C.Comp$ ⟨*proof*⟩
sublocale Op_0: *underlying-category* C T α ι *Obj* $Op.Hom_{op}$ *Id* $Op.Comp_{op}$ ⟨*proof*⟩
sublocale UF: *underlying-functor* C T α ι
Obj $Op.Hom_{op}$ *Id* $Op.Comp_{op}$
‹*Collect ide*› *exp* $C.Id$ $C.Comp$
hom_o hom_a
⟨*proof*⟩

The following is Kelly's formula (1.32) for $Hom^{\leftarrow}\ f\ y : Hom\ b\ y \rightarrow Hom\ a\ y$.

lemma *Kelly-1-32*:
assumes $a \in Obj$ **and** $b \in Obj$ **and** «$f : \mathcal{I} \rightarrow Hom\ a\ b$»
shows $C.frmUC\ (UF.map_0\ (Op_0.MkArr\ b\ a\ f)) =$
$Comp\ a\ b\ y \cdot (Hom\ b\ y \otimes f) \cdot \mathrm{r}^{-1}[hom_o\ b]$
⟨*proof*⟩

abbreviation map_0
where $map_0\ a\ b\ f \equiv Comp\ a\ b\ y \cdot (Hom\ b\ y \otimes f) \cdot \mathrm{r}^{-1}[hom_o\ b]$

end

context *elementary-closed-symmetric-monoidal-category*
begin

interpretation *enriched-category C T α ι ‹Collect ide› exp Id Comp*
⟨*proof*⟩
interpretation *self-enriched-category C T α ι exp eval Curry* ⟨*proof*⟩

sublocale *Op*: *opposite-enriched-category C T α ι σ ‹Collect ide› exp Id Comp*
⟨*proof*⟩

lemma *cnt-Exp-DN*:
assumes «$f : \mathcal{I} \rightarrow exp\ a\ b$»
and *ide a* **and** *ide b* **and** *ide y*
shows $Exp^{\leftarrow}\ (f\ ^{\downarrow}[a,\ b])\ y =$
$(Curry[exp\ a\ b,\ exp\ b\ y,\ exp\ a\ y]\ (Op.Comp_{op}\ y\ b\ a) \cdot f)$
$^{\downarrow}[exp\ b\ y,\ exp\ a\ y]$
⟨*proof*⟩

end

2.5 Enriched Yoneda Lemma

In this section we prove the (weak) Yoneda lemma for enriched categories, as in Kelly, Section 1.9. The weakness is due to the fact that the lemma asserts only a bijection between sets, rather than an isomorphism of objects of the underlying base category.

2.5.1 Preliminaries

The following gives conditions under which τ defined as $\tau\ x = (\mathcal{T}\ x)^{\uparrow}$ yields an enriched natural transformation between enriched functors F and G to the self-enriched base category.

context *elementary-closed-monoidal-category*
begin

lemma *transformation-lam-UP*:
assumes *enriched-functor C T α ι*
$Obj_A\ Hom_A\ Id_A\ Comp_A\ (Collect\ ide)\ exp\ Id\ Comp\ F_o\ F_a$
assumes *enriched-functor C T α ι*
$Obj_A\ Hom_A\ Id_A\ Comp_A\ (Collect\ ide)\ exp\ Id\ Comp\ G_o\ G_a$
and $\bigwedge x.\ x \notin Obj_A \Longrightarrow \mathcal{T}\ x = null$
and $\bigwedge x.\ x \in Obj_A \Longrightarrow$ «$\mathcal{T}\ x : F_o\ x \rightarrow G_o\ x$»

and $\bigwedge a\ b.\ [\![a \in Obj_A;\ b \in Obj_A]\!] \Longrightarrow$
$\mathcal{T}\ b \cdot Uncurry[F_o\ a,\ F_o\ b]\ (F_a\ a\ b) =$
$eval\ (G_o\ a)\ (G_o\ b) \cdot (G_a\ a\ b \otimes \mathcal{T}\ a)$
shows *enriched-natural-transformation* $C\ T\ \alpha\ \iota$
$Obj_A\ Hom_A\ Id_A\ Comp_A\ (Collect\ ide)\ exp\ Id\ Comp$
$F_o\ F_a\ G_o\ G_a\ (\lambda x.\ (\mathcal{T}\ x)^{\uparrow})$
⟨*proof*⟩

end

Kelly (1.39) expresses enriched naturality in an alternate form, using the underlying functors of the covariant and contravariant enriched hom functors.

locale *Kelly-1-39* =
symmetric-monoidal-category +
elementary-closed-monoidal-category +
enriched-natural-transformation
for a :: $'a$
and b :: $'a$ +
assumes a: $a \in Obj_A$
and b: $b \in Obj_A$
begin

interpretation *enriched-category* $C\ T\ \alpha\ \iota$ ‹*Collect ide*› *exp Id Comp*
⟨*proof*⟩
interpretation *self-enriched-category* $C\ T\ \alpha\ \iota$ *exp eval Curry*
⟨*proof*⟩

sublocale *cov-Hom*: *covariant-Hom* $C\ T\ \alpha\ \iota$
exp eval Curry $Obj_B\ Hom_B\ Id_B\ Comp_B$ ‹$F_o\ a$›
⟨*proof*⟩
sublocale *cnt-Hom*: *contravariant-Hom* $C\ T\ \alpha\ \iota\ \sigma$
exp eval Curry $Obj_B\ Hom_B\ Id_B\ Comp_B$ ‹$G_o\ b$›
⟨*proof*⟩

lemma *Kelly-1-39*:
shows $cov\text{-}Hom.map_0\ (F_o\ b)\ (G_o\ b)\ (\tau\ b) \cdot F_a\ a\ b =$
$cnt\text{-}Hom.map_0\ (F_o\ a)\ (G_o\ a)\ (\tau\ a) \cdot G_a\ a\ b$
⟨*proof*⟩

end

2.5.2 Covariant Case

locale *covariant-yoneda-lemma* =
symmetric-monoidal-category +
C: *closed-symmetric-monoidal-category* +
covariant-Hom +
F: *enriched-functor* $C\ T\ \alpha\ \iota$ *Obj Hom Id Comp* ‹*Collect ide*› *exp C.Id C.Comp*

begin

interpretation C: *elementary-closed-symmetric-monoidal-category C T α ι σ exp eval Curry* ⟨*proof*⟩
interpretation C: *self-enriched-category C T α ι exp eval Curry* ⟨*proof*⟩

Every element $e : \mathcal{I} \to F_o\ x$ of $F_o\ x$ determines an enriched natural transformation τ_e: $hom\ x\ -\ \to F$. The formula here is Kelly (1.47): $\tau_e\ y$: $hom\ x\ y \to F\ y$ is obtained as the composite:

$$hom\ x\ y \xrightarrow{F_a\ x\ y} exp\ (F\ x)\ (F\ y) \xrightarrow{Exp^{\leftarrow}\ e\ (F\ y)} exp\ \mathcal{I}\ (F\ y) \longrightarrow F\ y$$

where the third component is a canonical isomorphism. This basically amounts to evaluating $F_a\ x\ y$ on element e of $F_o\ x$ to obtain an element of $F_o\ y$.

Note that the above composite gives an arrow $\tau_e\ y$: $hom\ x\ y \to F\ y$, whereas the definition of enriched natural transformation formally requires $\tau_e\ y$: $\mathcal{I} \to exp\ (hom\ x\ y)\ (F\ y)$. So we need to transform the composite to achieve that.

abbreviation *generated-transformation*
where *generated-transformation e* ≡
$\lambda y.\ (eval\ \mathcal{I}\ (F_o\ y) \cdot \mathrm{r}^{-1}[exp\ \mathcal{I}\ (F_o\ y)] \cdot Exp^{\leftarrow}\ e\ (F_o\ y) \cdot F_a\ x\ y)^{\uparrow}$

lemma *enriched-natural-transformation-generated-transformation*:
assumes «$e : \mathcal{I} \to F_o\ x$»
shows *enriched-natural-transformation C T α ι*
Obj Hom Id Comp (*Collect ide*) *exp C.Id C.Comp*
$hom_o\ hom_a\ F_o\ F_a$ (*generated-transformation e*)
⟨*proof*⟩

If τ: $hom\ x\ -\ \to F$ is an enriched natural transformation, then there exists an element $e_\tau : \mathcal{I} \to F\ x$ that generates τ via the preceding formula. The idea (Kelly 1.46) is to take:

$$e_\tau = \mathcal{I} \xrightarrow{Id\ x} hom_o\ x \xrightarrow{\tau\ x} F\ x$$

This amounts to the "evaluation of $\tau\ x$ at the identity on x".

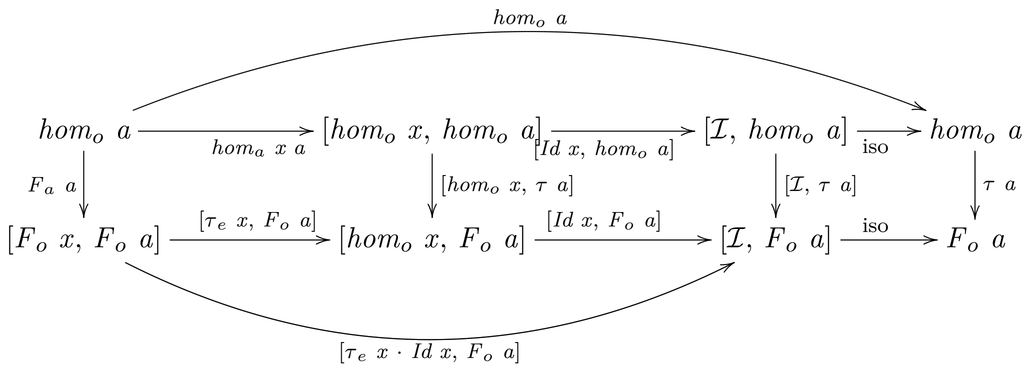
However, note once again that, according to the formal definition of enriched natural transformation, we have $\tau\ x : \mathcal{I} \to exp\ (hom_o\ x)\ (F_o\ x)$, so it is necessary to transform this to an arrow: $(\tau\ x)\ {}^{\downarrow}[hom_o\ x,\ F_o\ x]$: $hom_o\ x \to F\ x$.

abbreviation *generating-elem*
where *generating-elem* $\tau \equiv (\tau\ x)\ {}^{\downarrow}[hom_o\ x,\ F_o\ x] \cdot Id\ x$

lemma *generating-elem-in-hom*:
assumes *enriched-natural-transformation C T α ι*
Obj Hom Id Comp (*Collect ide*) *exp C.Id C.Comp*

hom$_o$ *hom*$_a$ F_o F_a τ
shows «*generating-elem* $\tau : \mathcal{I} \rightarrow F_o\ x$»
⟨*proof*⟩

Now we have to verify the elements of the diagram after Kelly (1.47):



The left square is enriched naturality of τ (Kelly (1.39)). The middle square commutes trivially. The right square commutes by the naturality of the canonical isomorphismm from $[\mathcal{I},\ hom_o\ a]$ to $hom_o\ a$. The top edge composes to $hom_o\ a$ (an identity). The commutativity of the entire diagram shows that $\tau\ a$ is recovered from e_τ. Note that where $\tau\ a$ appears, what is actually meant formally is $(\tau\ a)\ {}^{\downarrow}[hom_o\ a,\ F_o\ a]$.

lemma *center-square*:
assumes *enriched-natural-transformation* C T α ι
Obj Hom Id Comp (*Collect ide*) *exp* $C.Id$ $C.Comp$
hom$_o$ *hom*$_a$ F_o F_a τ
and $a \in Obj$
shows $C.Exp\ \mathcal{I}\ (\tau\ a\ {}^{\downarrow}[hom_o\ a,\ F_o\ a]) \cdot C.Exp\ (Id\ x)\ (hom_o\ a) =$
$C.Exp\ (Id\ x)\ (F_o\ a) \cdot C.Exp\ (hom_o\ x)\ (\tau\ a\ {}^{\downarrow}[hom_o\ a,\ F_o\ a])$
⟨*proof*⟩

lemma *right-square*:
assumes *enriched-natural-transformation* C T α ι
Obj Hom Id Comp (*Collect ide*) *exp* $C.Id$ $C.Comp$
hom$_o$ *hom*$_a$ F_o F_a τ
and $a \in Obj$
shows $\tau\ a\ {}^{\downarrow}[hom_o\ a,\ F_o\ a] \cdot C.Dn\ (hom_o\ a) =$
$C.Dn\ (F_o\ a) \cdot C.Exp\ \mathcal{I}\ (\tau\ a\ {}^{\downarrow}[hom_o\ a,\ F_o\ a])$
⟨*proof*⟩

lemma *top-path*:
assumes $a \in Obj$

shows *eval* $\mathcal{I}$ (hom_o *a*) · r^{-1}[*exp* $\mathcal{I}$ (hom_o *a*)] · *C.Exp* (*Id x*) (hom_o *a*) ·
hom_a *x a* =
hom_o *a*
⟨*proof*⟩

The left square is an instance of Kelly (1.39), so we can get that by instantiating that result. The confusing business is that the target enriched category is the base category C.

lemma *left-square*:
assumes *enriched-natural-transformation C T* α ι
Obj Hom Id Comp (*Collect ide*) *exp C.Id C.Comp*
hom_o hom_a F_o F_a τ
and $a \in Obj$
shows $Exp^{\rightarrow}$ (hom_o *x*) ((τ *a*) $^{\downarrow}$[hom_o *a*, F_o *a*]) · hom_a *x a* =
$Exp^{\leftarrow}$ ((τ *x*) $^{\downarrow}$[hom_o *x*, F_o *x*]) (F_o *a*) · F_a *x a*
⟨*proof*⟩

lemma *transformation-generated-by-element*:
assumes *enriched-natural-transformation C T* α ι
Obj Hom Id Comp (*Collect ide*) *exp C.Id C.Comp*
hom_o hom_a F_o F_a τ
and $a \in Obj$
shows τ *a* = *generated-transformation* (*generating-elem* τ) *a*
⟨*proof*⟩

lemma *element-of-generated-transformation*:
assumes $e \in hom\ \mathcal{I}\ (F_o\ x)$
shows *generating-elem* (*generated-transformation e*) = *e*
⟨*proof*⟩

We can now state and prove the (weak) covariant Yoneda lemma (Kelly, Section 1.9) for enriched categories.

theorem *covariant-yoneda*:
shows *bij-betw generated-transformation*
(*hom* $\mathcal{I}$ (F_o *x*))
(*Collect* (*enriched-natural-transformation C T* α ι
Obj Hom Id Comp (*Collect ide*) *exp C.Id C.Comp*
hom_o hom_a F_o F_a))
⟨*proof*⟩

end

2.5.3 Contravariant Case

The (weak) contravariant Yoneda lemma is obtained by just replacing the enriched category by its opposite in the covariant version.

locale *contravariant-yoneda-lemma* =
opposite-enriched-category C T α ι σ *Obj Hom Id Comp* +

covariant-yoneda-lemma C T α ι σ *exp eval Curry Obj* Hom_{op} *Id* $Comp_{op}$ y F_o F_a
for C :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixr** ‹·› *55*)
and T :: $'a \times 'a \Rightarrow 'a$
and α :: $'a \times 'a \times 'a \Rightarrow 'a$
and ι :: $'a$
and σ :: $'a \times 'a \Rightarrow 'a$
and *exp* :: $'a \Rightarrow 'a \Rightarrow 'a$
and *eval* :: $'a \Rightarrow 'a \Rightarrow 'a$
and *Curry* :: $'a \Rightarrow 'a \Rightarrow 'a \Rightarrow 'a \Rightarrow 'a$
and *Obj* :: $'b$ *set*
and *Hom* :: $'b \Rightarrow 'b \Rightarrow 'a$
and *Id* :: $'b \Rightarrow 'a$
and *Comp* :: $'b \Rightarrow 'b \Rightarrow 'b \Rightarrow 'a$
and y :: $'b$
and F_o :: $'b \Rightarrow 'a$
and F_a :: $'b \Rightarrow 'b \Rightarrow 'a$
begin

corollary *contravariant-yoneda*:
shows *bij-betw generated-transformation*
(*hom* $\mathcal{I}$ (F_o *y*))
(*Collect*
(*enriched-natural-transformation*
C T α ι *Obj* Hom_{op} *Id* $Comp_{op}$ (*Collect ide*) *exp C.Id C.Comp*
hom_o hom_a F_o F_a))
⟨*proof*⟩

end

end

Bibliography

[1] G. M. Kelly. Basic concepts of enriched category theory. *Reprints in Theory and Applications of Categories*, 10, 2005. http://www.tac.mta.ca/tac/reprints/articles/10/tr10.pdf.

[2] nLab. internal hom. *nLab (various contributors)*, 2009 – 2024. https://ncatlab.org/nlab/show/internal+hom, [Online; accessed 22-May-2024].

[3] E. W. Stark. Monoidal categories. *Archive of Formal Proofs*, May 2017. https://isa-afp.org/entries/MonoidalCategory.html, Formal proof development.

[4] E. W. Stark. Residuated transition systems II: Categorical properties. *Archive of Formal Proofs*, June 2024. (submitted for publication).