Electronic Notes in Theoretical Computer Science 40 (2001)

URL: <http://www.elsevier.nl/locate/entcs/volume40.html> 16 pages

A graphical approach to monad compositions

P. Eklund a M.A. Gal´an a*,*1 J. Medina b M. Ojeda-Aciego b

A. Valverde b

a *Ume˚a University, Department of Computing Science, SE-901 87 Ume˚a, Sweden*

b *Universidad de Ma´laga, Dept Matem´atica Aplicada, E-29071 Ma´laga, Spain* 2

**Abstract**

In this paper we show how composite expressions involving natural transformations can be pictorially represented in order to provide graphical proof support for pro- viding monad compositions. Examples are drawn using powerset monads composed with the term monad.

# Introduction

Monads have shown to be useful in different fields related to computer science. In functional programming monad compositions are applied to structuring of functional programs [11]. In particular, in functional programs like parsers or type checkers the monad needed is often a composed monad [13]. In logic programming, unification has been identified as the provision of co-equalisers in Kleisli categories of term monads [12].

The foundational understanding of monads has been well-known for decades, but proof techniques, especially related to monad compositions have not been developed. As monad compositions are basically built upon operations of corresponding natural transformations, proof techniques require an adequate handling of the basic combinatorial properties of functors and natural trans- formations (Godement rules). In [4,7] it was discovered that these combinato- rial properties can be represented more visually, in that the basic observation relates to distributivity of the star product of natural transformations with respect to composition of natural transformations.

This improves readability of expressions involving compositions of natural transformations and supports proofs involving more complex properties. This visual technique is not widely known and has been used mainly in purely algebraic contexts [1].

1 Supported by the Swedish Research Council for Engineering Sciences.

2 Partially supported by Spanish DGI project BFM2000-1054-C02-02

*◯*c *2001 Published by Elsevier Science B. V.* Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

The aim of this paper is to further develop these ideas about graphical representation, and to demonstrate the use of this technique on a case study for providing some concrete examples on generalised terms where various set functors are composed with the conventional term functor [6,5].

The structure of the paper is the following: in Section 2, the basic defi- nitions and notation of the graphical approach are given. In Section 3, some conditions for extending the composition of monads to a monad are presented and proved using the graphical approach. As an example, in Section 4 we introduce the problem of generalising sets of terms as the composition of suit- able powerset monads and the term monads, which raise our interest in the problem of composing monads and in a visual interface to make the corre- sponding calculations. Some conclusions are presented in Section 5. Finally, an appendix is introduced containing a standard proof of Proposition 3.2, just to see how the properties of natural transformations are more naturally han- dled in the graphical approach, so that one can abstract them from the main line of reasoning.

# Notations and pictorial representations

The notational conventions followed in this paper are those presented in [2]. Let C be a category and consider (covariant) endofunctors, denoted with capi- tal letters *F, G, H,...* : C *→* C, together with natural transformations, denoted with greek letters *τ, σ,.. .*, between such endofunctors. For *τ* : *F → G* and *σ*: *G → H*, let *σ ◦ τ* : *F → H* be the usual vertical composition of natural transformations, and for *τ '*: *F ' → G'*, let *τ ' \* τ* : *F ' ◦ F → G' ◦ G* be the star product given by

* 1. *τ ' \*τ* = *τ 'G ◦ F 'τ* = *G'τ ◦ τ 'F.*

The star product, like composition, is associative.

For the identity transformation *idF* : *F → F* , also written as 1*F* or 1, note that

* 1. 1*F \** 1*G* = 1*F◦G.*

For a natural transformation *τ* : *F → G*, and a functor *H*, (*Hτ* )*X* = *HτX* and (*τH*)*X* = *τHX*, or equivalently, *Hτ* = 1*H \*τ* and *τH* = *τ \** 1*H* . The following distributivity laws hold:

* 1. 1 *\** (*σ ◦ τ* )= (1 *\* σ*) *◦* (1 *\* τ* )*,*
  2. (*σ ◦ τ* ) *\** 1= (*σ \** 1) *◦* (*τ \** 1)*.*

A natural transformation *τ* : *F → G* as a basic building block is depicted as

*F*

*τ*

*G*

.

Blocks *τ* : *F → G* and *σ*: *G → H* are built, or composed, vertically as

*F F*

*σ*

*H*

*τ*

*G*

*σ ◦ τ H*

= .

For *τ* : *F → G* and *τ '*: *F ' → G'*, the horizontal composition of *τ '* followed by *τ* , denoted by the star product *τ ' \*τ* is visually denoted by the juxtaposition of two building blocks.

*F ' F*

*τ G*

*τ '*

*G'*

=

*F ' F*

.

*τ ' \*τ*

*G'*

*G*

Note in particular that the juxtaposition order reflects the syntactic order of *τ ' \* τ* . As an application of the previous construction, note that equation

(1) can be pictorially represented by

*F ' F*

*F ' F*

= =

*τ ' \*τ*

*G'*

*G*

*G*

*G'*

*τ ' \** 1*G*

*G*

*F '*

1*F' \*τ*

*F ' F*

.

*G*

*G'*

1*G' \*τ*

*F*

*G'*

*τ ' \** 1*F*

Equation (3) can be written as

*K F K F*

*H*

*σ ◦ τ*

*K*

1*K*

1*K \*σ K H*

1*K \*τ K G*

= ,

i.e., in this case building blocks can be applied in any order. The same holds for equation (4).

For natural transformations *F →τ*

*G →σ*

*H* and

*' τ'*

*' σ'*

*'* we then

*F → G → H*

have

*F ' F*

*H*

*H'*

*σ' \*σ*

*G*

*G'*

*τ ' \*τ*

(1)

=

*F ' F*

(3)

=

*F ' F*

(1)

=

*F ' F*

(4)

=

*F ' F*

|  |
| --- |
| *τ ' \** 1*F*  *G' F* |
| 1*G' \*τ G' G* |
| 1*G' \*σ G' H* |
| *σ' \** 1*H*  *H' H* |

(1)

=

*F ' F*

|  |
| --- |
| *τ ' \** 1*F*  *G' F* |
| 1*G' \** (*σ ◦ τ* ) *G' H* |
| *σ' \** 1*H*  *H' H* |

,

*σ ◦ τ H*

*σ' ◦ τ ' H'*

i.e., we have (re)proved the Interchange Law

*H*

*H'*

1*H' \** (*σ ◦ τ* )

*F*

*H'*

(*σ' ◦ τ '*) *\** 1*F*

|  |
| --- |
| *τ ' \** 1*F*  *G' F* |
| *σ' \** 1*F*  *H' F* |
| 1*H' \** (*σ ◦ τ* ) *H' H* |

* 1. (*σ' ◦ τ '*) *\** (*σ ◦ τ* )= (*σ' \* σ*) *◦* (*τ ' \* τ* )

which can be summarized as

*F ' F*

|  |  |
| --- | --- |
| *τ '*  *G'* | *τ G* |
| *σ'*  *H'* | *σ H* |

=

*F ' F*

=

*H*

*σ ◦ τ*

*H'*

*σ' ◦ τ '*

*F ' F*

showing how blocks with particular positions generally can be attached ver- tically and horizontally in any order without changing the resulting transfor- mation.

*H*

*H'*

*σ' \*σ*

*G*

*G'*

*τ ' \*τ*

Note in the transformation

*F F F*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *τ*  *F* | | *F* | | *σ*  *F* | |
| *F* | *σ* | | *F* | | *τ*  *F* |

that the composition (*σ\* τ* ) *◦* (*τ \* σ*) indeed exists, but neither *τ ◦ σ* nor *σ ◦ τ* do. This indicates how the applicability of the Interchange Law is more easily seen in the pictorial representation of the transformation.

In order to further improve readability of transformation expressions, iden- tity transformations 1*F* : *F −→ F* as blocks within transformation expressions are depicted as

*F*

or or

*F*  *F*

This choice for the representation of identity transformations will allow the use of asymmetric stacking of boxes.

# Monad compositions

A *monad* (or *triple*, or *algebraic theory*) over C is written as **F** = (*F, η, µ*), where *F* : C *→* C is a (covariant) functor, and *η*: *id*C *→ F* and *µ*: *F ◦ F → F* are natural transformations such that

1. *µ ◦* (*η \** 1*F* )= 1*F ,*
2. *µ ◦* (1*F \* η*)= 1*F ,*
3. *µ ◦* (1*F \* µ*)= *µ ◦* (*µ\** 1*F* )*.*

We say that *η* is respectively a left and right unit, and that the multiplication *µ* is associative. These monad conditions, with the identity functor *id*C: C *→* C written as 1, can be depicted as

1 *F F*

*µ F*

*η*

*F*

1*F*

*F*

*F* 1 *F*

*F F F F F F*

= ,

= ,

= .

The following proposition appears in [5]. Similar results concerning com- posability of monads appeared in [4,6,9], and originally also in [3].

*µ F*

*η*

*F*

*µ F*

*µ*

*F*

*µ F*

*µ*

*F*

1*F*

*F*

**Proposition 3.1** *Let* **F** = (*F, ηF , µF* ) *and* **G** = (*G, ηG, µG*) *be monads. Let σ*: *G ◦ F → F ◦ G, called a ‘swapper’, be a natural transformation such that the following properties hold:*

1 *F*

*F* 1

=

*ηG G*

1*F*

*F*

*σ*

*F G*

*ηF F*

*G* 1

*σ*

*F G*

*ηG G*

1 *G*

1*G*

*G*

*ηF F*

=

*G F G F*

=

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *F* | *σ* | | *G* |  |  | |  |
|  | | *µG G* | | | | |
|  |  | *F* | | *σ* | *G* |
| *µF F* | | | | | | |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | *F* | *σ* | | *G* |
| *µF F* | | | | |  | |
| *F* | *σ* | *G* | |  |  |
|  | *µG G* | | | | | | |

*G F G F*

*Then* **F** *•* **G** = (*F ◦ G, ηFG, µFG*) *is a monad, where*

1. *ηFG* = *ηF \* ηG,*
2. *µFG* = (*µF \* µG*) *◦* (1*F \*σ \** 1*G*)*.*

**Proof.** The following proof demonstrates the use of our pictorial representa- tions.

Firstly, we show that *ηFG* is a left unit.

1 *F G*

*µFG F G*

*ηFG F G*

(12)*,*(13)

=

1 1 *F G*

(9)

=

1 *F* 1 *G*

(6)

=

*F G*

|  |  |  |  |
| --- | --- | --- | --- |
| *ηF F* | *ηG G* |  |  |
|  | *σ*  *F G* | |
| *µF F* | | *µG G* | |

(2)

1*F \** 1*G*

*F G*

=

*FG*

.

1*FG*

*FG*

Note how the ‘highlighting’ of subexpressions is due to the Interchange Law. The right unit property is shown similarly.

|  |  |  |  |
| --- | --- | --- | --- |
| *ηF F* |  | *ηG G* |  |
| *µF F* | | *µG G* | |

Secondly, we show that *µFG* is associative.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *σ*  *F G* | |  |  | |
| *µF F* | | *µG G* | |
|  | | *σ*  *F G* | | |  |
| *µF F* | | | | *µG G* | |

*F G F G F G*

|  |  |  |  |
| --- | --- | --- | --- |
| *F* | *µFG* | *G* |  |
| *F* | | *µFG* | *G* |

(13)

=

*F G F G F G*

(1)

=

(1)

=

*F G F G F G*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *σ*  *F G* | |  |  | |
| *µF F* | | *µG G* | |
|  | |
|  | | *σ*  *F G* | | |  |
| *µF F* | | | | *µG G* | |

*F G F G F G*

*µG*

*G*

*µF*

*F*

*µF*

*F*

*µG*

*G*

*G*

*F*

*σ*

*σ*

*F G*

*F G F G F G*

*µG*

*G*

*µF F*

*µF*

*F*

*µG*

*G*

*G*

*F*

*σ*

*σ*

*F G*

(5)

=

(5)

=

*F G F G F G*

*µG*

*G*

*µF*

*F*

*σ*

*F*

*G*

*µF µG*

*F G*

*σ*

*F G*

*F G F G F G*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *σ*  *F G* | |  |  | |
| *µF F* | | *µG G* | |
| *σ*  *F G* | | |  |
|  | | *µG G* | |
| *µF F* | | | |

*F G F G F G*

*µG*

*G*

*µF F*

*µF*

*F*

*µG*

*G*

*G*

*F*

*σ*

*σ*

*F G*

(8)

=

(5)

=

*F G F G F G*

*µF*

*F*

*µF*

*F*

*µG G*

*µG*

*G*

*G*

*F*

*σ*

*σ*

*F G*

*F G F G F G*

|  |  |  |  |
| --- | --- | --- | --- |
|  | *F* | *µFG* | *G* |
| *F* | *µFG* | *G* | |

(11)

=

*re*=*vert* 3 *✷*

A converse result can be partially achieved under some additional assump- tions on the behaviour of the multiplication of the composite monad w.r.t. ei- ther the multiplications or the units of the base monads.

**Proposition 3.2** *If* **F** *•* **G** = (*F ◦ G, ηF \* ηG, µ*) *is a monad, then a natural transformation σµ*: *G ◦ F → F ◦ G can be deﬁned by*

1. *σµ* = *µ ◦* (*ηF \** 1*GF \* ηG*)

*such that conditions (9) and (10) are satisﬁed. In addition, condition (11) holds and µ* = *µFG, with µFG related to σµ given by (13), under the assumption that at least one of the conditions*

*F F G F G*

=

|  |  |  |
| --- | --- | --- |
|  | *µ*  *F G* | |
| *µF F* | | 1*G*  *G* |

*F* 1 *G F G*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1*F*  *F* | *ηF F* |  | | |
|  | *F* | *µ* | *G* |

=

*F F G F G*

*µ*

*F G*

*µF F*

1 *F G F G*

*µ*

*F G*

*ηF F*

*hold together with at least one of the conditions*

3 Simply apply the same steps in reverse ordering.

*F G F G G*

=

|  |  |  |
| --- | --- | --- |
| *µ*  *F G* | |  |
| 1*F*  *F* | *µG G* | |

*F G F* 1 *G*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | *ηG G* | 1*G*  *G* |
| *F* | *µ* | *G* |  |

=

*F G F G G*

*µ*

*F G*

*µG G*

*F G F G* 1

*ηG G*

*µ*

*F G*

**Proof.** Condition (9) follows from

1 *F*

*σµ*

*F G*

*ηG G*

(14)

=

1 1 *F* 1

=

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *ηF F* | *ηG G* | |  | | *ηG G* |
|  | |
|  | *F* | *µ* | | *G* |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *ηF \* ηG F G* | |  | | *ηG G* |
|  |
| *F* | *µ* | | *G* |  |

1 *F* 1

(6)

=

*F* 1

(2)

1*FG*

*F G*

*ηG G*

=

*F* 1

,

*ηG G*

1*F*

*F*

and condition (10) can be shown similarly.

Now, required combinations of conditions (15)-(18) imply condition (11), as shown by

1 *G F* 1 *G F* 1

*µ*

*F*

*G*

*µF*

*F*

1

*µ*

*F*

*G*

*µG*

*G*

1*F*

*F*

*ηG G*

*ηG G*

*ηF F*

*ηF F*

*G F G F*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | |  | *F* | *σµ* | | *G* |
| *µF F* | | | | |  | |
| *F* | *σµ* | | *G* | |  |  |
| 1*F*  *F* | | *µG G* | | | | | | |

(14)

=

1 *G F* 1 *G F* 1

1*F*

1 *F*

*ηF F*

*ηG G*

*µ*

*F*

*G*

*µG*

*G*

*µ*

*F G*

*ηG G*

1

*F*

*µF*

*ηF F*

1 *G F G F* 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *ηF F* |  | |  | | | | | *ηG G* |
|  | | |  |  | *F* | *µ* | *G* |  |
|  | *F* | *µ* | | *G* |  |  |  |  |

(16)*,*(17)

=

(6)

=

*G F G F*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *F* | *σµ* | | *G* |  |  | |  | |
|  | | *µG G* | | | | |
|  |  | *F* | | *σµ* | | *G* |
| *µF F* | | | | | | | 1*G*  *G* | |

1 *G F G F* 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *ηF F* |  | | | | |  | | *ηG G* |
|  | *F* | *µ* | *G* |  |  |  | | |
|  |  |  |  | *F* | *µ* | | *G* |  |

(8)

=

*re*=*vert* .

This provides a proof sketch. Other combinations of conditions can easily be applied in a similar way, also likewise in order to prove *µ* = *µFG*. *✷*

# 4 Examples of monad compositions

In this section we refer to some examples of monads and swappers that provide monad compositions. This example is interesting in that sufficient conditions for a composition of monads to be a monad were obtained using the graphical approach introduced above, and it illustrates how the complexity of checking monad conditions can be reduced to checking only a few sufficient conditions. Let *L* be a completely distributive lattice. For *L* = *{*0*,* 1*}*, write *L* = 2.

The covariant powerset functor *Lid* is obtained by *LidX* = *LX*, i.e., the set of

mappings *A*: *X → L*, and following [8], for a morphism *f* : *X → Y* in Set, by defining

*Lidf* (*A*)(*y*)=

*f* (*x*)=*y*

*A*(*x*)*.*

Further, define *η*:1 *→ Lid* by

*ηX*

(*x*)(*x'*)= 1 if *x* = *x'*

0 otherwise

and *µ*: *Lid ◦ Lid → Lid* by

*µX*(M)(*x*)=

*A∈LidX*

*A*(*x*) *∧* M(*A*)*.*

Then, **L***id* = (*Lid, η, µ*) is a monad [10], and **2***id* is the usual covariant powerset monad **P** = (*P, η, µ*), where *PX* is the set of subsets of *X*, *ηX*(*x*)= *{x}* and *µX*(B)= B.

These powerset monads are suitably composed with the term monad **T**Ω = (*T*Ω*, ηT*Ω *, µT*Ω ) [10], where *T*Ω*X* is the usual set of terms over an operator

domain Ω and variables in *X*, i.e., *T*Ω*X* = *∞ Tk*(*X*), where *T* 0(*X*) = *X*

*k*=0

Ω

Ω

and *Tk*+1(*X*) = *{*(*n, ω,* (*mi*)*i≤n*) *| ω ∈* Ω*n,n ∈ N, mi ∈ Tk*(*X*)*}*. In [6],

Ω Ω

a swapper *σ*: *T*Ω *◦ Lid → Lid ◦ T*Ω was given by *σX|T* 0*LX* = (1*Lid* )*X* and for

*l* = (*n, ω,* (*li*)*i≤n*) *∈ TαLX*, *α >* 0, *li ∈ Tβi LX*, *βi < α*, by

 *i≤n σX*(*li*)(*mi*) if *n* = *n'* and *ω* = *ω'*,

*σX*(*l*)((*n', ω',* (*mi*)*i≤n*)) =

 0 otherwise,

and it was shown that **L***id* and **T**Ω together with *σ* satisfy conditions in Propo- sitions 3.1 and 3.2.

# 5 Conclusions

The impact of the paper is two-fold. On one hand, we contribute to meth- ods and tools for generating monad compositions. In particular, we focus on composing various powerset monads with the term monad in order to pro- vide generalised terms for extended many-valued logic programming. On the other hand, we provide a categorical instrumentation for unification in the framework of using generalised terms.

We have shown how compositions and star products of natural transfor- mations can be pictorially represented in order to provide proof support. Han- dling conditions for monad compositions involve manipulations of rather com- plicated expressions involving natural transformations, and it is important to continue investigations on how to construct new monads from given ones.

Proving composability conditions is complicated as the complexity of the functors increase. The graphical support is beneficial in that composability proofs are expected to reveal further examples of monad compositions that provide useful scenarios for generalised terms. Not only is the graphical ap- proach a theoretical tool for a better understanding of the composition of natural transformations, but computing with natural transformations could be, to some extent, automatised and managed with such a graphical interface.

# References

1. Barja P´erez, J., *Personal communication*, april, 2000.
2. Barr, M. and C. Wells, “Category Theory for Computing Science,” Prentice Hall, 1990.
3. Beck, J., *Distributive laws*, in: *Seminar on Triples and Categorical Homology Theory*, number 80 in Lecture Notes in Mathematics, Springer-Verlag, 1969 pp. 119–130.
4. Caruncho Castro, J., *Triples theory*, Technical Report 5, Department of Algebra of the Univ. de Santiago de Compostela (1971), in Spanish.
5. Eklund, P., M. A. Gala´n, J. Medina, M. Ojeda-Aciego and A. Valverde, *Generalised terms and composition of monads*, in: *Proc. of ESTYLF 2000, Spanish Congress on Fuzzy Logic and Technology*, 2000, pp. 155–160.
6. Eklund, P., M. A. Gala´n, M. Ojeda-Aciego and A. Valverde, *Set functors and generalised terms*, in: *Proc. of IPMU 2000*, Information Processing and Management of Uncertainty in Knowledge-based Systems **vol. III**, 2000, pp. 1595–1599.
7. Freire Nistal, J., *Universal properties in higher order triples*, Technical Report 11, Department of Algebra of the Univ. de Santiago de Compostela (1971), in Spanish.
8. Goguen, J., *L-fuzzy sets*, J. Math. Anal. Appl. **18** (1967), pp. 145–174.
9. Jones, M. P. and L. Duponcheel, *Composing monads*, YALEU/DCS/RR-1004 Tech. Rep., Yale University (1993), available at the following web address <http://www.cse.ogi.edu/> mpj/pubs/composing.html.
10. Manes, E., “Algebraic Theories,” Springer-Verlag, 1976.
11. Moggi, E., *Notions of computation and monads*, Information and Computation

**93** (1991), pp. 55–92.

1. Rydeheard, D. and R. Burstall, *A categorical uniﬁcation algorithm*, in: *Proc. Category Theory and Computer Programming* (1986), pp. 493–505.
2. Wadler, P., *Comprehending monads*, Mathematical Structures in Computer Science **2** (1992), pp. 461–493.

# 6 Appendix: A standard proof of Proposition 3.2

For comparison purposes, we present here a standard proof of the result stated in Proposition 3.2. Firstly, the statement in rephrased in more conventional terms; then, the proof itself is given.

**Proposition 3.2** *If* **F** *•* **G** = (*F ◦ G, ηF G ◦ ηG, µ*) *is a monad, then a natural transformation σµ*: *G ◦ F → F ◦ G can be deﬁned by σµ* = *µ ◦ F GF ηG ◦ ηF GF such that conditions (9) and (10) are satisﬁed. In addition, condition (11) holds and µ* = *µ*(*σµ*) *under the assumption of any pair of properties* (*Ai, Bj*) *with i, j ∈ {*1*,* 2*}, where*

***(****A*1***)*** *µF G ◦ Fµ* = *µ ◦ µF GF G*

***(****A*2***)*** *Fµ ◦ FηF GF G* = *Fµ ◦ ηF F GFG*

***(****B*1***)*** *FµG ◦ µG* = *µ ◦ F GF µG*

***(****B*2***)*** *µG ◦ F GF ηGG* = *µG ◦ F GF GηG*

**Proof.**

Condition (9) follows from the fact that the unit transformation of the

composed monad is the composition of the units of *F* and *G*.

*σµ ◦ ηGF* = *µ ◦ F GF ηG ◦ ηF GF ◦ ηGF*

= *µ ◦ F GF ηG ◦ ηFGF*

= *µ ◦ ηFGFG ◦ FηG*

= *idFG ◦ FηG*

= *FηG.*

and condition (10) can be shown similarly.

Unfolding the definition of *σµ*, the left hand side (LHS) of equation (11) can be written as

*µF G ◦ Fµ ◦ FηF GF G ◦ F GF ηG ◦ FµGF ◦ µGF ◦ F GF ηGGF ◦ ηF GF GF*

and the right hand side (RHS) as

*F µG ◦ µG ◦ F GF ηGG ◦ ηF GF G ◦ GµF G ◦ GF µ ◦ GF ηF GF G ◦ GF GF ηG*

Assume, for instance, the properties (*A*2*, B*1).

For the LHS of equation (11) we have

*µF G ◦ Fµ ◦ FηF GF G ◦ F GF ηG ◦ FµGF ◦ µGF ◦ F GF ηGGF ◦ ηF GF GF A*2*,B*1

=

*µF G ◦ Fµ ◦ ηF F GFG ◦ F GF ηG ◦ µF ◦ F GF µGF ◦ F GF ηGGF ◦ ηF GF GF* =

= *µ ◦ µF GF G ◦ ηF F GFG ◦ F GF ηG ◦ µF ◦ ηF GF GF*

= *µ ◦ F GF ηG ◦ µF ◦ ηF GF GF.*

For the RHS we have

*F µG ◦ µG ◦ F GF ηGG ◦ ηF GF G ◦ GµF G ◦ GF µ ◦ GF ηF GF G ◦ GF GF ηG B*1*,A*2

=

*µ ◦ F GF µG ◦ F GF ηGG ◦ ηF GF G ◦ GµF G ◦ GF µ ◦ GηF F GFG ◦ GF GF ηG*

= *µ ◦ ηF GF G ◦ Gµ ◦ GµF GF G ◦ GηF F GFG ◦ GF GF ηG*

= *µ ◦ ηF GF G ◦ Gµ ◦ GF GF ηG.*

Note that, in fact, assuming any of the combinations (*Ai, Bj*), it can be proved that the LHS of equation (11) is equivalent to

*µ ◦ F GF ηG ◦ µF ◦ ηF GF GF*

whereas the RHS is equivalent to

*µ ◦ ηF GF G ◦ Gµ ◦ GF GF ηG*

The equivalence of the two previous equations follows directly from the graphical representation, although at the level of equational representation it cannot be easily observed. The proof follows as a consequence of the properties of natural transformations, as shown below:

*µ ◦ F GF ηG ◦ µF ◦ ηF GF GF* = *µ ◦ µFG ◦ F GF GF ηG ◦ ηF GF GF*

= *µ ◦ F Gµ ◦ ηF GF GF G ◦ GF GF ηG*

= *µ ◦ ηF GF G ◦ Gµ ◦ GF GF ηG.*

*✷*