Contents

1 Monoidal Category

1.1 Symmetric Monoidal Category

Definition 1. *A symmetric monoidal catergory (SMC),* $(\mathbb{C}, \bullet, 1, \alpha, \lambda, \rho, \gamma)$*, is a category* $\mathbb{C}$ *equippped with a bifunctor* $\bullet : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ *with a neutral element 1 and natural isomorphisms* α, λ, ρ*, and* γ*:*

1. $\alpha_{A,B,C} : A \bullet (B \bullet C) \xrightarrow{\sim} (A \bullet B) \bullet C$
2. $\lambda_A : 1 \bullet A \xrightarrow{\sim} A$
3. $\rho_A : A \bullet 1 \xrightarrow{\sim} A$
4. $\gamma_{A,B} : A \bullet B \xrightarrow{\sim} B \bullet A$

which make the following 'coherence' diagrams commute.

$$diagramshere$$

The following equality is also require to hold:

$$\lambda_1 \quad = \quad \rho_1 : 1 \bullet 1 \to 1$$

1.2 Symmetric Monoidal Closed Category

Definition 2. *A symmetric monoidal closed category (SMCC),* $(\mathbb{C}, \bullet, \multimap, 1, \alpha, \lambda, \rho, \gamma)$*, is a SMC such that for all objects* A *in* $\mathbb{C}$*, the functor* $- \otimes A$ *has a specified right adjoint* $A \multimap -$*.*

Let $\mathbb{C}$ *be a SMC* $(\mathbb{C}, \bullet, 1, \alpha, \lambda, \rho, \gamma)$*. A structure ,* M *, in* $\mathbb{C}$ *for a given signature* S_g *is specified by giving an object* $[\![\sigma]\!]$ *in* $\mathbb{C}$ *for each type* σ*, and a morphism* $[\![f]\!] : [\![\sigma_1]\!] \bullet ... \bullet [\![\sigma_n]\!] \to [\![\tau]\!]$ *in* $\mathbb{C}$ *for each function symbol* $f : \sigma_1, ..., \sigma_n \to \tau$*. In the case where* $n = 0$ *then the structure assigns a morphism* $[\![c]\!] : 1 \to [\![\tau]\!]$ *to a constant* $c : \tau$*.*

Given a context $\Gamma = [x_1 : \sigma_1, ..., x_n : \sigma_n]$ *we define* $[\![\Gamma]\!]$ *to be the product* $[\![\sigma_1]\!] \bullet ... \bullet [\![\sigma_n]\!]$*. We represent the empyt context with the neutral element 1. We need to define the bracketing convention. It shall be assumed that the tensor product is left associative, i.e.* $A_1 \bullet A_2 \bullet ... \bullet A_n$ *will be taken to mean* $(...(A_1 \bullet A_2) \bullet ...) \bullet A_n$*. We find it useful to define two 'book-keeping' functions,*

$$Split(\Gamma, \Delta) : [\![\Gamma, \Delta]\!] \to [\![\Gamma]\!] \bullet [\![\Delta]\!]$$

$$Split(\Gamma, \Delta) \stackrel{\mathsf{def}}{=} \begin{cases} \lambda_\Delta^{-1} & \textit{If } \Gamma \textit{ empty} \\ \rho_\Gamma^{-1} & \textit{If } \Delta \textit{ empty} \\ id_{\Gamma \bullet A} & \textit{If } \Delta = A \\ Split(\Gamma, \Delta') \bullet id_A ; \alpha^{-1}_{\Gamma, \Delta', A} & \textit{If } \Delta = \Delta', A \end{cases}$$

$$Join(\Gamma, \Delta) : [\![\Gamma, \Delta]\!] \to [\![\Gamma]\!] \bullet [\![\Delta]\!]$$

$$Join(\Gamma, \Delta) \stackrel{\mathsf{def}}{=} \begin{cases} \lambda_\Delta & \textit{If } \Gamma \textit{ empty} \\ \rho_\Gamma & \textit{If } \Delta \textit{ empty} \\ id_{\Gamma \bullet A} & \textit{If } \Delta = A \\ \alpha_{\Gamma, \Delta', A} ; Join(\Gamma, \Delta') \bullet id_A & \textit{If } \Delta = \Delta', A \end{cases}$$

We shall also refer to indexed variants of these; for example

$$Split_n(\Gamma_1, ..., \Gamma_n) : [\![\Gamma_1, ..., \Gamma_n]\!] \to [\![\Gamma_1]\!] \bullet ... \bullet [\![\Gamma_n]\!]$$

which is defined in the obvious way.

The semantics of a term in context is then specified by a structural induction on the term.

$$[\![x : \sigma \rhd x : \sigma]\!] \stackrel{\mathsf{def}}{=} id_\sigma$$

$$[\![\Gamma_1, ..., \Gamma_n \rhd f(M_1, ..., M_n) : \tau]\!] \stackrel{\mathsf{def}}{=} Split_n(\Gamma_1, ..., \Gamma_n); [\![\Gamma_1 \rhd M_1 : \sigma_1]\!] \bullet ... \bullet [\![\Gamma_n \rhd M_1 : \sigma_n 0]\!]; [\![f]\!]$$

2 Monodial Funtor

3 Monodial Adjuctions