DiracDec C++/Coq Implementation[1]

December 3, 2024

1 Language Syntax

Definition 1.1 (syntax)**.** *The syntax for Dirac notation types is defined as*

$$T ::= x \mid \mathcal{S} \mid \mathcal{K}(\sigma) \mid \mathcal{B}(\sigma) \mid \mathcal{O}(\sigma_1, \sigma_2) \mid T_1 \to T_2,$$
$$\sigma ::= x \mid \sigma_1 \times \sigma_2.$$

The syntax for Dirac notation terms is defined as

$$\begin{aligned} e ::= {} & x \mid (e_1, e_2) \\ & \mid 0 \mid 1 \mid \mathsf{ADDS}(e_1 \cdots e_n) \mid e_1 \times \cdots \times e_n \mid e^* \mid \delta_{e_1,e_2} \mid \mathsf{DOT}(e_1\ e_2) \\ & \mid \mathbf{0}_{\mathcal{K}}(\sigma) \mid \mathbf{0}_{\mathcal{B}}(\sigma) \mid \mathbf{0}_{\mathcal{O}}(\sigma_1, \sigma_2) \mid \mathbf{1}_{\mathcal{O}}(\sigma) \\ & \mid |e\rangle \mid \langle t| \mid e^\dagger \mid e_1.e_2 \mid \mathsf{ADD}(e_1 \cdots e_n) \mid e_1 \otimes e_2 \\ & \mid \mathsf{MULK}(e_1\ e_2) \mid \mathsf{MULB}(e_1\ e_2) \mid \mathsf{OUTER}(e_1\ e_2) \mid \mathsf{MULO}(e_1\ e_2). \end{aligned}$$

Compared to [?], this syntax for Dirac notations merges the symbols with overlapped properties, such as the addition and scaling symbols for ket, bra and operator. Here ADDS and ADD are two different AC symbols representing the scalar addition and the linear algebra addition respectively. They will be denoted as $a_1 + \cdots + a_n$ and $X_1 + \cdots + X_n$. There are five kinds of linear algebra multiplications among ket, bra and operator, whose properties are similar but still diverge to some extent. For example, the rules $(O_1 \cdot O_2) \cdot K \ \triangleright\ O_1 \cdot (O_2 \cdot K)$ and $B \cdot (O_1 \cdot O_2) \triangleright (B \cdot O_1) \cdot O_2$ indicate that the sorting of multiplication sequences depends on the subterm types. To avoid frequent but unnecessary type checkings, we encode the typing information by using five different symbols, namely DOT, MULK, MULB, OUTER and MULO. They are denoted as $B \cdot K$, $K_1 \cdot K_2$, $B_1 \cdot B_2$, $K \cdot B$ and $O_1 \cdot O_2$, respectively.

2 Typing System

The type checking of Dirac notations involves maintaining a well-formed context Γ, which specifies the definitions and typing assumtpions for variables. Notice that the type checking $\Gamma \vdash t : T$ is decidable.

$$\textbf{W-Empty}\quad \frac{}{\mathcal{WF}([])} \qquad \textbf{W-Assum-Type}\quad \frac{\mathcal{WF}(\Gamma) \qquad T \notin \Gamma}{\mathcal{WF}(\Gamma; T : \mathsf{Type})}$$

$$\textbf{W-Assum-Term}\quad \frac{\Gamma \vdash T : \mathsf{Type} \qquad x \notin \Gamma}{\mathcal{WF}(\Gamma; x : T)} \qquad \textbf{W-Def}\quad \frac{\Gamma \vdash t : T \qquad x \notin \Gamma}{\mathcal{WF}(\Gamma; x := t : T)}$$

$$\textbf{Var}\quad \frac{\mathcal{WF}(\Gamma) \qquad (x : T) \in \Gamma \text{ or } (x := t : T) \in \Gamma \text{ for some } t}{\Gamma \vdash x : T}$$

$$\textbf{Arrow}\quad \frac{\Gamma \vdash T : \mathsf{Type} \qquad \Gamma \vdash U : \mathsf{Type}}{\Gamma \vdash T \to U : \mathsf{Type}}$$

$$\textbf{Lam}\quad \frac{\Gamma; x : T \vdash t : U}{\Gamma \vdash (\lambda x : T.t) : T \to U} \qquad \textbf{App}\quad \frac{\Gamma \vdash t : U \to T \qquad \Gamma \vdash u : U}{\Gamma \vdash (t\ u) : T}$$

Figure 1: The typing rules for the typed lambda calculus with environment.

$$\textbf{Type-Base}\quad \frac{\mathcal{WF}(\Gamma)}{\Gamma \vdash \mathsf{Base} : \mathsf{Type}} \qquad \textbf{Base-Subtype}\quad \frac{\Gamma \vdash T : \mathsf{Base}}{\Gamma \vdash T : \mathsf{Type}}$$

$$\textbf{Type-Ket}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base}}{\Gamma \vdash \mathcal{K}(\sigma) : \mathsf{Type}} \qquad \textbf{Type-Bra}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base}}{\Gamma \vdash \mathcal{B}(\sigma) : \mathsf{Type}}$$

$$\textbf{Type-Opt}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base} \qquad \Gamma \vdash \tau : \mathsf{Base}}{\Gamma \vdash \mathcal{O}(\sigma, \tau) : \mathsf{Type}} \qquad \textbf{Type-Scalar}\quad \frac{\mathcal{WF}(\Gamma)}{\Gamma \vdash \mathcal{S} : \mathsf{Type}}$$

Figure 2: Typing rules for Dirac notation types.

$$\textbf{Sca-0}\quad \frac{\mathcal{WF}(\Gamma)}{\Gamma \vdash 0 : \mathcal{S}} \qquad \textbf{Sca-1}\quad \frac{\mathcal{WF}(\Gamma)}{\Gamma \vdash 1 : \mathcal{S}} \qquad \textbf{Sca-Delta}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base} \quad \Gamma \vdash s : \sigma \qquad \Gamma \vdash t : \sigma}{\Gamma \vdash \delta_{s,t} : \mathcal{S}}$$

$$\textbf{Sca-Add}\quad \frac{\Gamma \vdash a_i : \mathcal{S} \text{ for all } i}{\Gamma \vdash a_1 + \cdots + a_n : \mathcal{S}} \qquad \textbf{Sca-Mul}\quad \frac{\Gamma \vdash a_i : \mathcal{S} \text{ for all } i}{\Gamma \vdash a_1 \times \cdots \times a_n : \mathcal{S}}$$

$$\textbf{Sca-Conj}\quad \frac{\Gamma \vdash a : \mathcal{S}}{\Gamma \vdash a^* : \mathcal{S}} \qquad \textbf{Sca-Dot}\quad \frac{\Gamma \vdash B : \mathcal{B}(\sigma) \qquad \Gamma \vdash K : \mathcal{K}(\sigma)}{\Gamma \vdash B \cdot K : \mathcal{S}}$$

Figure 3: Scalar typing rules.

$$\textbf{Base-Prod}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base} \qquad \Gamma \vdash \tau : \mathsf{Base}}{\Gamma \vdash \sigma \times \tau : \mathsf{Base}} \qquad \textbf{Pair-Base}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base} \quad \Gamma \vdash s : \sigma \qquad \Gamma \vdash \tau : \mathsf{Base} \quad \Gamma \vdash t : \tau}{\Gamma \vdash (s, t) : \sigma \times \tau}$$

Figure 4: Typing rules for bases.

$$\textbf{Ket-0}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base}}{\Gamma \vdash \mathbf{0}_{\mathcal{K}}(\sigma) : \mathcal{K}(\sigma)} \qquad \textbf{Ket-Base}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base} \qquad \Gamma \vdash t : \sigma}{\Gamma \vdash |t\rangle : \mathcal{K}(\sigma)}$$

$$\textbf{Ket-Adj}\quad \frac{\Gamma \vdash B : \mathcal{B}(\sigma)}{\Gamma \vdash B^\dagger : \mathcal{K}(\sigma)} \qquad \textbf{Ket-Scr}\quad \frac{\Gamma \vdash a : \mathcal{S} \qquad \Gamma \vdash K : \mathcal{K}(\sigma)}{\Gamma \vdash a.K : \mathcal{K}(\sigma)}$$

$$\textbf{Ket-Add}\quad \frac{\Gamma \vdash K_i : \mathcal{K}(\sigma) \text{ for all } i}{\Gamma \vdash K_1 + \cdots + K_n : \mathcal{K}(\sigma)} \qquad \textbf{Ket-MulK}\quad \frac{\Gamma \vdash O : \mathcal{O}(\sigma,\tau) \qquad \Gamma \vdash K : \mathcal{K}(\tau)}{\Gamma \vdash O \cdot K : \mathcal{K}(\sigma)}$$

$$\textbf{Ket-Tsr}\quad \frac{\Gamma \vdash K_1 : \mathcal{K}(\sigma) \qquad \Gamma \vdash K_2 : \mathcal{K}(\tau)}{\Gamma \vdash K_1 \otimes K_2 : \mathcal{K}(\sigma \times \tau)}$$

Figure 5: Ket typing rules.

$$\textbf{Bra-0}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base}}{\Gamma \vdash \mathbf{0}_{\mathcal{B}}(\sigma) : \mathcal{B}(\sigma)} \qquad \textbf{Bra-Base}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base} \qquad \Gamma \vdash t : \sigma}{\Gamma \vdash \langle t| : \mathcal{B}(\sigma)}$$

$$\textbf{Bra-Adj}\quad \frac{\Gamma \vdash K : \mathcal{K}(\sigma)}{\Gamma \vdash K^\dagger : \mathcal{B}(\sigma)} \qquad \textbf{Bra-Scr}\quad \frac{\Gamma \vdash a : \mathcal{S} \qquad \Gamma \vdash B : \mathcal{B}(\sigma)}{\Gamma \vdash a.B : \mathcal{B}(\sigma)}$$

$$\textbf{Bra-Add}\quad \frac{\Gamma \vdash B_i : \mathcal{B}(\sigma) \text{ for all } i}{\Gamma \vdash B_1 + \cdots + B_n : \mathcal{B}(\sigma)} \qquad \textbf{Bra-MulB}\quad \frac{\Gamma \vdash B : \mathcal{K}(\sigma) \qquad \Gamma \vdash O : \mathcal{O}(\sigma,\tau)}{\Gamma \vdash B \cdot O : \mathcal{B}(\tau)}$$

$$\textbf{Bra-Tsr}\quad \frac{\Gamma \vdash B_1 : \mathcal{B}(\sigma) \qquad \Gamma \vdash B_2 : \mathcal{B}(\tau)}{\Gamma \vdash B_1 \otimes B_2 : \mathcal{B}(\sigma \times \tau)}$$

Figure 6: Bra typing rules.

$$\textbf{Opt-0}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base} \qquad \Gamma \vdash \tau : \mathsf{Base}}{\Gamma \vdash \mathbf{0}_{\mathcal{O}}(\sigma,\tau) : \mathcal{O}(\sigma,\tau)} \qquad \textbf{Opt-1}\quad \frac{\Gamma \vdash \sigma : \mathsf{Base}}{\Gamma \vdash \mathbf{1}_{\mathcal{O}}(\sigma) : \mathcal{O}(\sigma,\sigma)}$$

$$\textbf{Opt-Adj}\quad \frac{\Gamma \vdash O : \mathcal{O}(\sigma,\tau)}{\Gamma \vdash O^\dagger : \mathcal{O}(\tau,\sigma)} \qquad \textbf{Opt-Scr}\quad \frac{\Gamma \vdash a : \mathcal{S} \qquad \Gamma \vdash O : \mathcal{O}(\sigma,\tau)}{\Gamma \vdash a.O : \mathcal{O}(\sigma,\tau)}$$

$$\textbf{Opt-Add}\quad \frac{\Gamma \vdash O_i : \mathcal{O}(\sigma,\tau) \text{ for all } i}{\Gamma \vdash O_1 + \cdots + O_n : \mathcal{O}(\sigma,\tau)} \qquad \textbf{Opt-Outer}\quad \frac{\Gamma \vdash K : \mathcal{K}(\sigma) \qquad \Gamma \vdash B : \mathcal{B}(\tau)}{\Gamma \vdash K \cdot B : \mathcal{O}(\sigma,\tau)}$$

$$\textbf{Opt-MulO}\quad \frac{\Gamma \vdash O_1 : \mathcal{O}(\sigma,\tau) \qquad \Gamma \vdash O_2 : \mathcal{O}(\tau,\rho)}{\Gamma \vdash O_1 \cdot O_2 : \mathcal{O}(\sigma,\rho)}$$

$$\textbf{Opt-Tsr}\quad \frac{\Gamma \vdash O_1 : \mathcal{O}(\sigma_1,\tau_1) \qquad \Gamma \vdash O_2 : \mathcal{O}(\sigma_2,\tau_2)}{\Gamma \vdash O_1 \otimes O_2 : \mathcal{O}(\sigma_1 \times \sigma_2, \tau_1 \times \tau_2)}$$

Figure 7: Operator typing rules.

3 Conversions and Reductions

Alpha (bound variable renaming) **Beta** $\dfrac{}{\Gamma \vdash ((\lambda x : T.t)\ u)\ \triangleright_\beta\ t\{x/u\}}$

Delta $\dfrac{\mathcal{WF}(\Gamma) \qquad (c := t : T) \in \Gamma}{\Gamma \vdash c\ \triangleright_\delta\ t}$ **Eta** $\dfrac{\Gamma \vdash t : T \to U}{\Gamma \vdash \lambda x : T.(t\ x)\ \triangleright_\eta\ t}$

Figure 8: Conversions and reductions for the typed lambda calculus.

[YX] : Alpha rule will be implemented by renaming, and we can leave it for the future.

Table 1: The special flattening rule.

Rule	Description
R-FLATTEN	$a_1 + \cdots + (b_1 + \cdots + b_m) + \cdots + a_n \ \triangleright\ a_1 + \cdots + b_1 + \cdots + b_m + \cdots + a_n$
	$a_1 \times \cdots \times (b_1 \times \cdots \times b_m) \times \cdots \times a_n \ \triangleright\ a_1 \times \cdots \times b_1 \times \cdots \times b_m \times \cdots \times a_n$
	$X_1 + \cdots + (X'_1 + \cdots + X'_m) + \cdots + X_n \ \triangleright\ X_1 + \cdots + X'_1 + \cdots + X'_m + \cdots + X_n$
	A special rule to flatten all AC symbols within one call.

Table 2: Rules for scalar addition and multiplication.

Rule	Description
R-ADDSID	$+(a) \ \triangleright\ a$
	Reduce the term if the AC symbol + only has one argument.
R-MULSID	$\times(a) \ \triangleright\ a$
	Similar to (R-ADDSID).
R-ADDS0	$a_1 + \cdots + 0 + \cdots + a_n \ \triangleright\ a_1 + \cdots + a_n$
	This rule removes all 0 occurances and keeps the order of remaining subterms.
R-MULS0	$a_1 \times \cdots \times 0 \times \cdots \times a_n \ \triangleright\ 0$
R-MULS1	$a_1 \times \cdots \times 1 \times \cdots \times a_n \ \triangleright\ a_1 \times \cdots \times a_n$
	Similar to (R-ADDS0).
R-MULS2	$b_1 \times \cdots \times (a_1 + \cdots + a_n) \times \cdots \times b_m$
	$\triangleright (b_1 \times \cdots \times a_1 \times \cdots \times b_m) + \cdots + (b_1 \times \cdots \times a_n \times \cdots \times b_m)$
	This rule matches the first scalar addition subterm in the list.

Table 3: Rules for other scalar symbols.

Rule	Description
R-CONJ0	$0^* \ \triangleright\ 0$
R-CONJ1	$1^* \ \triangleright\ 1$
R-CONJ2	$(a_1 + \cdots + a_n)^* \ \triangleright\ a_1^* + \cdots + a_n^*$
R-CONJ3	$(a_1 \times \cdots \times a_n)^* \ \triangleright\ a_1^* \times \cdots \times a_n^*$
R-CONJ4	$(a^*)^* \ \triangleright\ a$
R-CONJ5	$\delta_{s,t}^* \ \triangleright\ \delta_{s,t}$

Continued on the next page

Rule	Description
R-CONJ6	$(B \cdot K)^* \ \triangleright \ K^\dagger \cdot B^\dagger$
R-DOT0	$\mathbf{0}_{\mathcal{B}}(\sigma) \cdot K \ \triangleright \ 0$
R-DOT1	$B \cdot \mathbf{0}_{\mathcal{K}}(\sigma) \ \triangleright \ 0$
R-DOT2	$(a.B) \cdot K \ \triangleright \ a \times (B \cdot K)$
R-DOT3	$B \cdot (a.K) \ \triangleright \ a \times (B \cdot K)$
R-DOT4	$(B_1 + \cdots + B_n) \cdot K \ \triangleright \ B_1 \cdot K + \cdots + B_n \cdot K$
R-DOT5	$B \cdot (K_1 + \cdots + K_n) \ \triangleright \ B \cdot K_1 + \cdots + B \cdot K_n$
R-DOT6	$\langle s| \cdot |t\rangle \ \triangleright \ \delta_{s,t}$
R-DOT7	$(B_1 \otimes B_2) \cdot |(s,t)\rangle \ \triangleright \ (B_1 \cdot |s\rangle) \times (B_2 \cdot |t\rangle)$
R-DOT8	$\langle (s,t)| \cdot (K_1 \otimes K_2) \ \triangleright \ (\langle s| \cdot K_1) \times (\langle t| \cdot K_2)$
R-DOT9	$(B_1 \otimes B_2) \cdot (K_1 \otimes K_2) \ \triangleright \ (B_1 \cdot K_1) \times (B_2 \cdot K_2)$
R-DOT10	$(B \cdot O) \cdot K \ \triangleright \ B \cdot (O \cdot K)$
R-DOT11	$\langle (s,t)| \cdot ((O_1 \otimes O_2) \cdot K) \ \triangleright \ ((\langle s| \cdot O_1) \otimes (\langle t| \cdot O_2)) \cdot K$
R-DOT12	$(B_1 \otimes B_2) \cdot ((O_1 \otimes O_2) \cdot K) \ \triangleright \ ((B_1 \cdot O_1) \otimes (B_2 \cdot O_2)) \cdot K$
R-DELTA0	$\delta_{a,a} \ \triangleright \ 1$
R-DELTA1	$\delta_{(a,b),(c,d)} \ \triangleright \ \delta_{a,c} \times \delta_{b,d}$

Table 4: Rules for linear algebra scaling.

Rule	Description
R-SCR0	$1.X \ \triangleright \ X$
R-SCR1	$a.(b.X) \ \triangleright \ (a \times b).X$
R-SCR2	$a.(X_1 + \cdots + X_n) \ \triangleright \ a.X_1 + \cdots + a.X_n$
R-SCRK0	$K : \mathcal{K}(\sigma) \Rightarrow 0.K \ \triangleright \ \mathbf{0}_{\mathcal{K}}(\sigma)$
R-SCRK1	$a.\mathbf{0}_{\mathcal{K}}(\sigma) \ \triangleright \ \mathbf{0}_{\mathcal{K}}(\sigma)$
R-SCRB0	$B : \mathcal{B}(\sigma) \Rightarrow 0.B \ \triangleright \ \mathbf{0}_{\mathcal{B}}(\sigma)$
R-SCRB1	$a.\mathbf{0}_{\mathcal{B}}(\sigma) \ \triangleright \ \mathbf{0}_{\mathcal{B}}(\sigma)$
R-SCRO0	$O : \mathcal{O}(\sigma,\tau) \Rightarrow 0.O \ \triangleright \ \mathbf{0}_{\mathcal{O}}(\sigma,\tau)$
R-SCRO1	$a.\mathbf{0}_{\mathcal{O}}(\sigma,\tau) \ \triangleright \ \mathbf{0}_{\mathcal{O}}(\sigma,\tau)$

Table 5: Rules for linear algebra addition.

Rule	Description
R-ADDID	$+(X) \ \triangleright \ X$ Reduce the term if the AC symbol + only has one argument.
R-ADD0	$Y_1 + \cdots + X + \cdots + X + \cdots + Y_n \ \triangleright \ Y_1 + \cdots + Y_n + \cdots + (1+1).X$ This rule matches the first X in the list satisfying the pattern. The result $(1+1).X$ will be placed at the end of the list.
R-ADD1	$Y_1 + \cdots + X + \cdots + a.X + \cdots + Y_n \ \triangleright \ Y_1 + \cdots + Y_n + (1+a).X$ Similar to (R-ADD0).
R-ADD2	$Y_1 + \cdots + a.X + \cdots + X + \cdots + Y_n \ \triangleright \ Y_1 + \cdots + Y_n + (a+1).X$ Similar to (R-ADD0).
R-ADD3	$Y_1 + \cdots + a.X + \cdots + b.X + \cdots + Y_n \ \triangleright \ Y_1 + \cdots + Y_n + (a+b).X$ Similar to (R-ADD0).

Continued on the next page

Rule	Description
R-ADDK0	$K_1 + \cdots + \mathbf{0}_{\mathcal{K}}(\sigma) + \cdots + K_n \ \triangleright K_1 + \cdots + K_n$
	Similar to (R-ADDS0).
R-ADDB0	$B_1 + \cdots + \mathbf{0}_{\mathcal{B}}(\sigma) + \cdots + B_n \ \triangleright \ B_1 + \cdots + B_n$
	Similar to (R-ADDS0).
R-ADDO0	$O_1 + \cdots + \mathbf{0}_{\mathcal{O}}(\sigma,\tau) + \cdots + O_n \ \triangleright \ O_1 + \cdots + O_n$
	Similar to (R-ADDS0).

Table 6: Rules for adjoint.

Rule	Description
R-ADJ0	$(X^\dagger)^\dagger \ \triangleright \ X$
R-ADJ1	$(a.X)^\dagger \ \triangleright \ (a^*).(X^\dagger)$
R-ADJ2	$(X_1 + \cdots + X_n)^\dagger \ \triangleright \ X_1^\dagger + \cdots + X_n^\dagger$
R-ADJ3	$(X \otimes Y)^\dagger \ \triangleright \ X^\dagger \otimes Y^\dagger$
R-ADJK0	$\mathbf{0}_{\mathcal{B}}(\sigma)^\dagger \ \triangleright \ \mathbf{0}_{\mathcal{K}}(\sigma)$
R-ADJK1	$\langle t|^\dagger \ \triangleright \ |t\rangle$
R-ADJK2	$(B \cdot O)^\dagger \ \triangleright \ O^\dagger \cdot B^\dagger$
R-ADJB0	$\mathbf{0}_{\mathcal{K}}(\sigma)^\dagger \ \triangleright \ \mathbf{0}_{\mathcal{B}}(\sigma)$
R-ADJB1	$|t\rangle^\dagger \ \triangleright \ \langle t|$
R-ADJB2	$(O \cdot K)^\dagger \ \triangleright \ K^\dagger \cdot O^\dagger$
R-ADJO0	$\mathbf{0}_{\mathcal{O}}(\sigma,\tau)^\dagger \ \triangleright \ \mathbf{0}_{\mathcal{O}}(\tau,\sigma)$
R-ADJO1	$\mathbf{1}_{\mathcal{O}}(\sigma)^\dagger \ \triangleright \ \mathbf{1}_{\mathcal{O}}(\sigma)$
R-ADJO2	$(K \cdot B)^\dagger \ \triangleright \ B^\dagger \cdot K^\dagger$
R-ADJO3	$(O_1 \cdot O_2)^\dagger \ \triangleright \ O_2^\dagger \cdot O_1^\dagger$

Table 7: Rules for tensor product.

Rule	Description
R-TSR0	$(a.X_1) \otimes X_2 \ \triangleright \ a.(X_1 \otimes X_2)$
R-TSR1	$X_1 \otimes (a.X_2) \ \triangleright \ a.(X_1 \otimes X_2)$
R-TSR2	$(X_1 + \cdots + X_n) \otimes X' \triangleright X_1 \otimes X' + \cdots + X_n \otimes X'$
R-TSR3	$X' \otimes (X_1 + \cdots + X_n) \triangleright X' \otimes X_1 + \cdots + X' \otimes X_n$
R-TSRK0	$K : \mathcal{K}(\tau) \Rightarrow \mathbf{0}_{\mathcal{K}}(\sigma) \otimes K \ \triangleright \ \mathbf{0}_{\mathcal{K}}(\sigma \times \tau)$
R-TSRK1	$K : \mathcal{K}(\tau) \Rightarrow K \otimes \mathbf{0}_{\mathcal{K}}(\sigma) \ \triangleright \ \mathbf{0}_{\mathcal{K}}(\tau \times \sigma)$
R-TSRK2	$|s\rangle \otimes |t\rangle \ \triangleright \ |(s,t)\rangle$
R-TSRB0	$B : \mathcal{B}(\tau) \Rightarrow \mathbf{0}_{\mathcal{B}}(\sigma) \otimes B \ \triangleright \ \mathbf{0}_{\mathcal{B}}(\sigma \times \tau)$
R-TSRB1	$B : \mathcal{B}(\tau) \Rightarrow B \otimes \mathbf{0}_{\mathcal{B}}(\sigma) \ \triangleright \ \mathbf{0}_{\mathcal{B}}(\tau \times \sigma)$
R-TSRB2	$\langle s| \otimes \langle t| \ \triangleright \ \langle (s,t)|$
R-TSRO0	$O : \mathcal{O}(\sigma,\tau) \Rightarrow O \otimes \mathbf{0}_{\mathcal{O}}(\sigma',\tau') \ \triangleright \ \mathbf{0}_{\mathcal{O}}(\sigma \times \sigma', \tau \times \tau')$
R-TSRO1	$O : \mathcal{O}(\sigma,\tau) \Rightarrow \mathbf{0}_{\mathcal{O}}(\sigma',\tau') \otimes O \ \triangleright \ \mathbf{0}_{\mathcal{O}}(\sigma' \times \sigma, \tau' \times \tau)$
R-TSRO2	$\mathbf{1}_{\mathcal{O}}(\sigma) \otimes \mathbf{1}_{\mathcal{O}}(\tau) \ \triangleright \ \mathbf{1}_{\mathcal{O}}(\sigma \times \tau)$
R-TSRO3	$(K_1 \cdot B_1) \otimes (K_2 \cdot B_2) \ \triangleright \ (K_1 \otimes K_2) \cdot (B_1 \otimes B_2)$

Table 8: Rule for $O \cdot K$.

Rule	Description
R-MULK0	$\mathbf{0}_{\mathcal{O}}(\sigma,\tau) \cdot K \ \triangleright\ \mathbf{0}_{\mathcal{K}}(\sigma)$
R-MULK1	$O : \mathcal{O}(\sigma,\tau) \Rightarrow O \cdot \mathbf{0}_{\mathcal{K}}(\tau) \ \triangleright\ \mathbf{0}_{\mathcal{K}}(\sigma)$
R-MULK2	$\mathbf{1}_{\mathcal{O}}(\sigma) \cdot K \ \triangleright K$
R-MULK3	$(a.O) \cdot K \ \triangleright\ a.(O \cdot K)$
R-MULK4	$O \cdot (a.K) \ \triangleright\ a.(O \cdot K)$
R-MULK5	$(O_1 + \cdots + O_n) \cdot K \ \triangleright\ O_1 \cdot K + \cdots + O_n \cdot K$
R-MULK6	$O \cdot (K_1 + \cdots + K_n) \ \triangleright\ O \cdot K_1 + \cdots + O \cdot K_n$
R-MULK7	$(K_1 \cdot B) \cdot K_2 \ \triangleright\ (B \cdot K_2).K_1$
R-MULK8	$(O_1 \cdot O_2) \cdot K \ \triangleright\ O_1 \cdot (O_2 \cdot K)$
R-MULK9	$(O_1 \otimes O_2) \cdot ((O_1' \otimes O_2') \cdot K) \ \triangleright\ ((O_1 \cdot O_1') \otimes (O_2 \cdot O_2')) \cdot K$
R-MULK10	$(O_1 \otimes O_2) \cdot \lvert(s,t)\rangle \ \triangleright\ (O_1 \cdot \lvert s\rangle) \otimes (O_2 \cdot \lvert t\rangle)$
R-MULK11	$(O_1 \otimes O_2) \cdot (K_1 \otimes K_2) \ \triangleright\ (O_1 \cdot K_1) \otimes (O_2 \cdot K_2)$

Table 9: Rule for $B \cdot O$.

Rule	Description
R-MULB0	$B \cdot \mathbf{0}_{\mathcal{O}}(\sigma,\tau) \ \triangleright\ \mathbf{0}_{\mathcal{B}}(\tau)$
R-MULB1	$O : \mathcal{O}(\sigma,\tau) \Rightarrow \mathbf{0}_{\mathcal{B}}(\sigma) \cdot O \ \triangleright\ \mathbf{0}_{\mathcal{B}}(\tau)$
R-MULB2	$B \cdot \mathbf{1}_{\mathcal{O}}(\sigma) \ \triangleright B$
R-MULB3	$(a.B) \cdot O \ \triangleright\ a.(B \cdot O)$
R-MULB4	$B \cdot (a.O) \ \triangleright\ a.(B \cdot O)$
R-MULB5	$(B_1 + \cdots + B_n) \cdot O \ \triangleright\ B_1 \cdot O + \cdots + B_n \cdot O$
R-MULB6	$B \cdot (O_1 + \cdots + O_n) \ \triangleright\ B \cdot O_1 + \cdots + B \cdot O_n$
R-MULB7	$B_1 \cdot (K \cdot B_2) \ \triangleright\ (B_1 \cdot K).B_2$
R-MULB8	$B \cdot (O_1 \cdot O_2) \ \triangleright\ (B \cdot O_1) \cdot O_2$
R-MULB9	$(B \cdot (O_1' \otimes O_2')) \cdot (O_1 \otimes O_2) \ \triangleright\ B \cdot ((O_1' \otimes O_2') \cdot (O_1 \otimes O_2))$
R-MULB10	$\langle(s,t)\rvert \cdot (O_1 \otimes O_2) \ \triangleright\ (\langle s\rvert \cdot O_1) \otimes (\langle t\rvert \cdot O_2)$
R-MULB11	$(B_1 \otimes B_2) \cdot (O_1 \otimes O_2) \ \triangleright\ (B_1 \cdot O_1) \otimes (B_2 \cdot O_2)$

Table 10: Rules for $K \cdot B$.

Rule	Description
R-OUTER0	$B : \mathcal{B}(\tau) \Rightarrow \mathbf{0}_{\mathcal{K}}(\sigma) \cdot B \ \triangleright\ \mathbf{0}_{\mathcal{O}}(\sigma,\tau)$
R-OUTER1	$K : \mathcal{K}(\sigma) \Rightarrow K \cdot \mathbf{0}_{\mathcal{B}}(\tau) \ \triangleright\ \mathbf{0}_{\mathcal{O}}(\sigma,\tau)$
R-OUTER2	$(a.K) \cdot B \ \triangleright\ a.(K \cdot B)$
R-OUTER3	$K \cdot (a.B) \ \triangleright\ a.(K \cdot B)$
R-OUTER4	$(K_1 + \cdots + K_n) \cdot B \ \triangleright\ K_1 \cdot B + \cdots + K_n \cdot B$
R-OUTER5	$K \cdot (B_1 + \cdots + B_n) \ \triangleright\ K \cdot B_1 + \cdots + K \cdot B_n$

Table 11: Rules for $O_1 \cdot O_2$.

Rule	**Description**
R-MULO0	$O : \mathcal{O}(\tau, \rho) \Rightarrow \mathbf{0}_{\mathcal{O}}(\sigma, \tau) \cdot O \ \triangleright \ \mathbf{0}_{\mathcal{O}}(\sigma, \rho)$
R-MULO1	$O : \mathcal{O}(\sigma, \tau) \Rightarrow O \cdot \mathbf{0}_{\mathcal{O}}(\tau, \rho) \ \triangleright \ \mathbf{0}_{\mathcal{O}}(\sigma, \rho)$
R-MULO2	$\mathbf{1}_{\mathcal{O}}(\sigma) \cdot O \ \triangleright \ O$
R-MULO3	$O \cdot \mathbf{1}_{\mathcal{O}}(\sigma) \ \triangleright \ O$
R-MULO4	$(K \cdot B) \cdot O \ \triangleright \ K \cdot (B \cdot O)$
R-MULO5	$O \cdot (K \cdot B) \ \triangleright \ (O \cdot K) \cdot B$
R-MULO6	$(a.O_1) \cdot O_2 \ \triangleright \ a.(O_1 \cdot O_2)$
R-MULO7	$O_1 \cdot (a.O_2) \ \triangleright \ a.(O_1 \cdot O_2)$
R-MULO8	$(O_1 + \cdots + O_n) \cdot O' \ \triangleright \ O_1 \cdot O' + \cdots + O_n \cdot O'$
R-MULO9	$O' \cdot (O_1 + \cdots + O_n) \ \triangleright \ O' \cdot O_1 + \cdots + O' \cdot O_n$
R-MULO10	$(O_1 \cdot O_2) \cdot O_3 \ \triangleright \ O_1 \cdot (O_2 \cdot O_3)$
R-MULO11	$(O_1 \otimes O_2) \cdot (O_1' \otimes O_2') \ \triangleright \ (O_1 \cdot O_1') \otimes (O_2 \cdot O_2')$
R-MULO12	$(O_1 \otimes O_2) \cdot ((O_1' \otimes O_2') \cdot O_3) \ \triangleright \ ((O_1 \cdot O_1') \otimes (O_2 \cdot O_2')) \cdot O_3$

4 Diracoq language

$$
\begin{aligned}
cmd ::= & \texttt{Def}(\texttt{ID}\ term) \\
& \mid \texttt{Def}(\texttt{ID}\ term\ type) \\
& \mid \texttt{Var}(\texttt{ID}\ term) \\
& \mid \texttt{Check}(term) \\
& \mid \texttt{Show}(\texttt{ID}) \\
& \mid \texttt{ShowAll} \\
& \mid \texttt{Normalize}(term) \\
& \mid \texttt{CheckEq}(term\ term) \\
type ::= & \texttt{Type} \mid \texttt{Arrow}(type\ type) \\
& \mid \texttt{Base} \\
term ::= & \texttt{Type} \mid \texttt{fun}(\texttt{ID}\ type\ term) \mid \texttt{apply}(term\ term) \mid \texttt{ID}
\end{aligned}
$$

5 Rewriting Control and Intermediate Language

The associativity is already handled by the (R-FLATTEN) rule. In order to decide two terms A and B are equivalent under commutativity, we need to proof that A can be transformed into B with a structured permutation, which is described by the *permutation tree*.

Definition 5.1 (permutation tree)**.** *The syntax for permutation trees are inductively defined below:*

$$P ::= \mathsf{E} \mid [(i : P)^+].$$

Here i represents positive numbers.

We always only consider *well-formed* permutation trees. That is, if $P \equiv [i_1 : P_1\ i_2 : P_2\ \cdots\ i_n : P_n]$, then $\{i_1, ...i_n\}$ forms the set of integers from 0 to $n-1$.

We can transform a term A with a suitable permutation tree. The transformation is defined as

$$
\begin{aligned}
&\texttt{apply}(A, P) := \texttt{match}\ P\ \texttt{with} \\
&\qquad \mid \mathsf{E} \Rightarrow A \\
&\qquad \mid [i_1 : P_1\ \cdots\ i_n : P_n] \Rightarrow \texttt{A.head}(\texttt{apply}(\texttt{A.args}[i_1],\ P_{i_1})\ \cdots\ \texttt{apply}(\texttt{A.args}[i_n],\ P_{i_n})) \\
&\quad \texttt{end}
\end{aligned}
$$

References

[1] Walliam Shakespeard. Halmet. 2666.