

DiracDec C++/Coq Implementation[1]

December 5, 2024

1 TODO

- The Coq side should be able to decide the alpha-equivalence of terms.

2 Language Syntax

Definition 2.1 (syntax). *The syntax for Dirac notation types is defined as*

$$\begin{aligned} T &::= x \mid \mathcal{S} \mid \mathcal{K}(\sigma) \mid \mathcal{B}(\sigma) \mid \mathcal{O}(\sigma_1, \sigma_2) \mid T_1 \rightarrow T_2 \mid \text{Set}(\sigma), \\ \sigma &::= x \mid \sigma_1 \times \sigma_2. \end{aligned}$$

The syntax for Dirac notation terms is defined as

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

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.

Usually, the sum body is specified by an abstraction. Therefore we use notation $\sum_{i \in s} X$ to denote $\sum_s \lambda i. X$.

3 Typing System

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

W-Empty	$\frac{}{\mathcal{WF}(\square)}$	W-Assum-Type	$\frac{\mathcal{WF}(\Gamma) \quad T \notin \Gamma}{\mathcal{WF}(\Gamma; T : \text{Type})}$
W-Assum-Term	$\frac{\Gamma \vdash T : \text{Type} \quad x \notin \Gamma}{\mathcal{WF}(\Gamma; x : T)}$	W-Def	$\frac{\Gamma \vdash t : T \quad x \notin \Gamma}{\mathcal{WF}(\Gamma; x := t : T)}$
Var	$\frac{\mathcal{WF}(\Gamma) \quad (x : T) \in \Gamma \text{ or } (x := t : T) \in \Gamma \text{ for some } t}{\Gamma \vdash x : T}$		
Arrow	$\frac{\Gamma \vdash T : \text{Type} \quad \Gamma \vdash U : \text{Type}}{\Gamma \vdash T \rightarrow U : \text{Type}}$		
Lam	$\frac{\Gamma; x : T \vdash t : U}{\Gamma \vdash (\lambda x : T. t) : T \rightarrow U}$	App	$\frac{\Gamma \vdash t : U \rightarrow T \quad \Gamma \vdash u : U}{\Gamma \vdash (t \ u) : T}$

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

Type-Base	$\frac{\mathcal{WF}(\Gamma)}{\Gamma \vdash \text{Base} : \text{Type}}$	Base-Subtype	$\frac{\Gamma \vdash T : \text{Base}}{\Gamma \vdash T : \text{Type}}$
Type-Ket	$\frac{\Gamma \vdash \sigma : \text{Base}}{\Gamma \vdash \mathcal{K}(\sigma) : \text{Type}}$	Type-Bra	$\frac{\Gamma \vdash \sigma : \text{Base}}{\Gamma \vdash \mathcal{B}(\sigma) : \text{Type}}$
Type-Opt	$\frac{\Gamma \vdash \sigma : \text{Base} \quad \Gamma \vdash \tau : \text{Base}}{\Gamma \vdash \mathcal{O}(\sigma, \tau) : \text{Type}}$	Type-Scalar	$\frac{\mathcal{WF}(\Gamma)}{\Gamma \vdash \mathcal{S} : \text{Type}}$
Type-Set	$\frac{\Gamma \vdash \sigma : \text{Base}}{\Gamma \vdash \text{Set}(\sigma) : \text{Type}}$		

Figure 2: Typing rules for Dirac notation types.

Sca-0	$\frac{\mathcal{WF}(\Gamma)}{\Gamma \vdash 0 : \mathcal{S}}$	Sca-1	$\frac{\mathcal{WF}(\Gamma)}{\Gamma \vdash 1 : \mathcal{S}}$	Sca-Delta	$\frac{\Gamma \vdash \sigma : \text{Base} \quad \Gamma \vdash s : \sigma \quad \Gamma \vdash t : \sigma}{\Gamma \vdash \delta_{s,t} : \mathcal{S}}$
Sca-Add	$\frac{\Gamma \vdash a_i : \mathcal{S} \text{ for all } i}{\Gamma \vdash a_1 + \dots + a_n : \mathcal{S}}$	Sca-Mul	$\frac{\Gamma \vdash a_i : \mathcal{S} \text{ for all } i}{\Gamma \vdash a_1 \times \dots \times a_n : \mathcal{S}}$		
Sca-Conj	$\frac{\Gamma \vdash a : \mathcal{S}}{\Gamma \vdash a^* : \mathcal{S}}$	Sca-Dot	$\frac{\Gamma \vdash B : \mathcal{B}(\sigma) \quad \Gamma \vdash K : \mathcal{K}(\sigma)}{\Gamma \vdash B \cdot K : \mathcal{S}}$		

Figure 3: Scalar typing rules.

$$\begin{array}{c}
\textbf{Base-Prod} \quad \frac{\Gamma \vdash \sigma : \text{Base} \quad \Gamma \vdash \tau : \text{Base}}{\Gamma \vdash \sigma \times \tau : \text{Base}} \qquad \textbf{Pair-Base} \quad \frac{\Gamma \vdash \sigma : \text{Base} \quad \Gamma \vdash s : \sigma \quad \Gamma \vdash \tau : \text{Base} \quad \Gamma \vdash t : \tau}{\Gamma \vdash (s, t) : \sigma \times \tau}
\end{array}$$

Figure 4: Typing rules for bases.

$$\begin{array}{c}
\textbf{Ket-0} \quad \frac{\Gamma \vdash \sigma : \text{Base}}{\Gamma \vdash \mathbf{0}_{\mathcal{K}}(\sigma) : \mathcal{K}(\sigma)} \qquad \textbf{Ket-Base} \quad \frac{\Gamma \vdash \sigma : \text{Base} \quad \Gamma \vdash t : \sigma}{\Gamma \vdash |t\rangle : \mathcal{K}(\sigma)} \\
\textbf{Ket-Adj} \quad \frac{\Gamma \vdash B : \mathcal{B}(\sigma)}{\Gamma \vdash B^\dagger : \mathcal{K}(\sigma)} \qquad \textbf{Ket-Scr} \quad \frac{\Gamma \vdash a : \mathcal{S} \quad \Gamma \vdash K : \mathcal{K}(\sigma)}{\Gamma \vdash a.K : \mathcal{K}(\sigma)} \\
\textbf{Ket-Add} \quad \frac{\Gamma \vdash K_i : \mathcal{K}(\sigma) \text{ for all } i}{\Gamma \vdash K_1 + \dots + K_n : \mathcal{K}(\sigma)} \qquad \textbf{Ket-MulK} \quad \frac{\Gamma \vdash O : \mathcal{O}(\sigma, \tau) \quad \Gamma \vdash K : \mathcal{K}(\tau)}{\Gamma \vdash O \cdot K : \mathcal{K}(\sigma)} \\
\textbf{Ket-Tsr} \quad \frac{\Gamma \vdash K_1 : \mathcal{K}(\sigma) \quad \Gamma \vdash K_2 : \mathcal{K}(\tau)}{\Gamma \vdash K_1 \otimes K_2 : \mathcal{K}(\sigma \times \tau)}
\end{array}$$

Figure 5: Ket typing rules.

$$\begin{array}{c}
\textbf{Bra-0} \quad \frac{\Gamma \vdash \sigma : \text{Base}}{\Gamma \vdash \mathbf{0}_{\mathcal{B}}(\sigma) : \mathcal{B}(\sigma)} \qquad \textbf{Bra-Base} \quad \frac{\Gamma \vdash \sigma : \text{Base} \quad \Gamma \vdash t : \sigma}{\Gamma \vdash \langle t| : \mathcal{B}(\sigma)} \\
\textbf{Bra-Adj} \quad \frac{\Gamma \vdash K : \mathcal{K}(\sigma)}{\Gamma \vdash K^\dagger : \mathcal{B}(\sigma)} \qquad \textbf{Bra-Scr} \quad \frac{\Gamma \vdash a : \mathcal{S} \quad \Gamma \vdash B : \mathcal{B}(\sigma)}{\Gamma \vdash a.B : \mathcal{B}(\sigma)} \\
\textbf{Bra-Add} \quad \frac{\Gamma \vdash B_i : \mathcal{B}(\sigma) \text{ for all } i}{\Gamma \vdash B_1 + \dots + B_n : \mathcal{B}(\sigma)} \qquad \textbf{Bra-MulB} \quad \frac{\Gamma \vdash B : \mathcal{K}(\sigma) \quad \Gamma \vdash O : \mathcal{O}(\sigma, \tau)}{\Gamma \vdash B \cdot O : \mathcal{B}(\tau)} \\
\textbf{Bra-Tsr} \quad \frac{\Gamma \vdash B_1 : \mathcal{B}(\sigma) \quad \Gamma \vdash B_2 : \mathcal{B}(\tau)}{\Gamma \vdash B_1 \otimes B_2 : \mathcal{B}(\sigma \times \tau)}
\end{array}$$

Figure 6: Bra typing rules.

$$\begin{array}{c}
\textbf{Opt-0} \quad \frac{\Gamma \vdash \sigma : \text{Base} \quad \Gamma \vdash \tau : \text{Base}}{\Gamma \vdash \mathbf{0}_{\mathcal{O}}(\sigma, \tau) : \mathcal{O}(\sigma, \tau)} \qquad \textbf{Opt-1} \quad \frac{\Gamma \vdash \sigma : \text{Base}}{\Gamma \vdash \mathbf{1}_{\mathcal{O}}(\sigma) : \mathcal{O}(\sigma, \sigma)} \\
\\
\textbf{Opt-Adj} \quad \frac{\Gamma \vdash O : \mathcal{O}(\sigma, \tau)}{\Gamma \vdash O^\dagger : \mathcal{O}(\tau, \sigma)} \qquad \textbf{Opt-Scr} \quad \frac{\Gamma \vdash a : \mathcal{S} \quad \Gamma \vdash O : \mathcal{O}(\sigma, \tau)}{\Gamma \vdash a.O : \mathcal{O}(\sigma, \tau)} \\
\\
\textbf{Opt-Add} \quad \frac{\Gamma \vdash O_i : \mathcal{O}(\sigma, \tau) \text{ for all } i}{\Gamma \vdash O_1 + \dots + O_n : \mathcal{O}(\sigma, \tau)} \qquad \textbf{Opt-Outer} \quad \frac{\Gamma \vdash K : \mathcal{K}(\sigma) \quad \Gamma \vdash B : \mathcal{B}(\tau)}{\Gamma \vdash K \cdot B : \mathcal{O}(\sigma, \tau)} \\
\\
\textbf{Opt-MulO} \quad \frac{\Gamma \vdash O_1 : \mathcal{O}(\sigma, \tau) \quad \Gamma \vdash O_2 : \mathcal{O}(\tau, \rho)}{\Gamma \vdash O_1 \cdot O_2 : \mathcal{O}(\sigma, \rho)} \\
\\
\textbf{Opt-Tsr} \quad \frac{\Gamma \vdash O_1 : \mathcal{O}(\sigma_1, \tau_1) \quad \Gamma \vdash O_2 : \mathcal{O}(\sigma_2, \tau_2)}{\Gamma \vdash O_1 \otimes O_2 : \mathcal{O}(\sigma_1 \times \sigma_2, \tau_1 \times \tau_2)}
\end{array}$$

Figure 7: Operator typing rules.

$$\begin{array}{c}
\textbf{Set-U} \quad \frac{\Gamma \vdash \sigma : \text{Base}}{\Gamma \vdash \mathbf{U}(\sigma) : \text{Set}(\sigma)} \qquad \textbf{Set-Prod} \quad \frac{\Gamma \vdash A : \text{Set}(\sigma) \quad \Gamma \vdash B : \text{Set}(\tau)}{\Gamma \vdash A \star B : \text{Set}(\sigma \times \tau)}
\end{array}$$

Figure 8: Set typing rules.

$$\begin{array}{c}
\textbf{Sum-Scalar} \quad \frac{\Gamma \vdash s : \text{Set}(\sigma) \quad \Gamma \vdash f : \sigma \rightarrow \mathcal{S}}{\Gamma \vdash \sum_s f : \mathcal{S}} \qquad \textbf{Sum-Ket} \quad \frac{\Gamma \vdash s : \text{Set}(\sigma) \quad \Gamma \vdash f : \sigma \rightarrow \mathcal{K}(\tau)}{\Gamma \vdash \sum_s f : \mathcal{K}(\tau)} \\
\\
\textbf{Sum-Bra} \quad \frac{\Gamma \vdash s : \text{Set}(\sigma) \quad \Gamma \vdash f : \sigma \rightarrow \mathcal{B}(\tau)}{\Gamma \vdash \sum_s f : \mathcal{B}(\tau)} \qquad \textbf{Sum-Opt} \quad \frac{\Gamma \vdash s : \text{Set}(\sigma) \quad \Gamma \vdash f : \sigma \rightarrow \mathcal{O}(\tau, \rho)}{\Gamma \vdash \sum_s f : \mathcal{O}(\tau, \rho)}
\end{array}$$

Figure 9: Sum typing rules.

4 Conversions and Reductions

Alpha	(bound variable renaming)	Beta	$\frac{}{\Gamma \vdash ((\lambda x : T.t) u) \triangleright_{\beta} t\{x/u\}}$
Delta	$\frac{\mathcal{WF}(\Gamma) \quad (c := t : T) \in \Gamma}{\Gamma \vdash c \triangleright_{\delta} t}$	Eta	$\frac{\Gamma \vdash t : T \rightarrow U}{\Gamma \vdash \lambda x : T.(t x) \triangleright_{\eta} t}$

Figure 10: 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	$a_1 + \dots + (b_1 + \dots + b_m) + \dots + a_n \triangleright a_1 + \dots + b_1 + \dots + b_m + \dots + a_n$ $a_1 \times \dots \times (b_1 \times \dots \times b_m) \times \dots \times a_n \triangleright a_1 \times \dots \times b_1 \times \dots \times b_m \times \dots \times a_n$ $X_1 + \dots + (X'_1 + \dots + X'_m) + \dots + X_n \triangleright X_1 + \dots + X'_1 + \dots + X'_m + \dots + X_n$ <p>A special rule to flatten all AC symbols within one call.</p>

Table 2: Rules for scalar addition and multiplication.

Rule	Description
R-ADDSD	$+(a) \triangleright a$ <p>Reduce the term if the AC symbol $+$ only has one argument.</p>
R-MULSD	$\times(a) \triangleright a$ <p>Similar to (R-ADDSD).</p>
R-ADDSD0	$a_1 + \dots + 0 + \dots + a_n \triangleright a_1 + \dots + a_n$ <p>This rule removes all 0 occurrences and keeps the order of remaining subterms.</p>
R-MULSD0	$a_1 \times \dots \times 0 \times \dots \times a_n \triangleright 0$
R-MULSD1	$a_1 \times \dots \times 1 \times \dots \times a_n \triangleright a_1 \times \dots \times a_n$ <p>Similar to (R-ADDSD0).</p>
R-MULSD2	$b_1 \times \dots \times (a_1 + \dots + a_n) \times \dots \times b_m$ $\triangleright (b_1 \times \dots \times a_1 \times \dots \times b_m) + \dots + (b_1 \times \dots \times a_n \times \dots \times b_m)$ <p>This rule matches the first scalar addition subterm in the list.</p>

Table 3: Rules for other scalar symbols.

Rule	Description
R-CONJ0	$0^* \triangleright 0$
R-CONJ1	$1^* \triangleright 1$
R-CONJ2	$(a_1 + \dots + a_n)^* \triangleright a_1^* + \dots + a_n^*$
R-CONJ3	$(a_1 \times \dots \times a_n)^* \triangleright a_1^* \times \dots \times a_n^*$
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	$(B_1 + \dots + B_n) \cdot K \triangleright B_1 \cdot K + \dots + B_n \cdot K$
R-DOT5	$B \cdot (K_1 + \dots + K_n) \triangleright B \cdot K_1 + \dots + B \cdot K_n$
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	$a.(X_1 + \dots + X_n) \triangleright a.X_1 + \dots + a.X_n$
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	$Y_1 + \dots + X + \dots + X + \dots + Y_n \triangleright Y_1 + \dots + Y_n + \dots + (1 + 1).X$ 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	$Y_1 + \dots + X + \dots + a.X + \dots + Y_n \triangleright Y_1 + \dots + Y_n + (1 + a).X$ Similar to (R-ADD0).
R-ADD2	$Y_1 + \dots + a.X + \dots + X + \dots + Y_n \triangleright Y_1 + \dots + Y_n + (a + 1).X$ Similar to (R-ADD0).
R-ADD3	$Y_1 + \dots + a.X + \dots + b.X + \dots + Y_n \triangleright Y_1 + \dots + Y_n + (a + b).X$ Similar to (R-ADD0).

Continued on the next page

Rule	Description
R-ADDK0	$K_1 + \dots + \mathbf{0}_{\mathcal{K}}(\sigma) + \dots + K_n \triangleright K_1 + \dots + K_n$ Similar to (R-ADDS0).
R-ADDB0	$B_1 + \dots + \mathbf{0}_{\mathcal{B}}(\sigma) + \dots + B_n \triangleright B_1 + \dots + B_n$ Similar to (R-ADDS0).
R-ADDO0	$O_1 + \dots + \mathbf{0}_{\mathcal{O}}(\sigma, \tau) + \dots + O_n \triangleright O_1 + \dots + O_n$ 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	$(X_1 + \dots + X_n)^\dagger \triangleright X_1^\dagger + \dots + X_n^\dagger$
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	$(X_1 + \dots + X_n) \otimes X' \triangleright X_1 \otimes X' + \dots + X_n \otimes X'$
R-TSR3	$X' \otimes (X_1 + \dots + X_n) \triangleright X' \otimes X_1 + \dots + X' \otimes X_n$
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	$(O_1 + \dots + O_n) \cdot K \triangleright O_1 \cdot K + \dots + O_n \cdot K$
R-MULK6	$O \cdot (K_1 + \dots + K_n) \triangleright O \cdot K_1 + \dots + O \cdot K_n$
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	$(O_1 \otimes O_2) \cdot (s, t)\rangle \triangleright (O_1 \cdot s\rangle) \otimes (O_2 \cdot t\rangle)$
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	$(B_1 + \dots + B_n) \cdot O \triangleright B_1 \cdot O + \dots + B_n \cdot O$
R-MULB6	$B \cdot (O_1 + \dots + O_n) \triangleright B \cdot O_1 + \dots + B \cdot O_n$
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	$\langle (s, t) \cdot (O_1 \otimes O_2) \triangleright (\langle s \cdot O_1) \otimes (\langle t \cdot O_2)$
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	$(K_1 + \dots + K_n) \cdot B \triangleright K_1 \cdot B + \dots + K_n \cdot B$
R-OUTER5	$K \cdot (B_1 + \dots + B_n) \triangleright K \cdot B_1 + \dots + K \cdot B_n$

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	$(O_1 + \dots + O_n) \cdot O' \triangleright O_1 \cdot O' + \dots + O_n \cdot O'$
R-MULO9	$O' \cdot (O_1 + \dots + O_n) \triangleright O' \cdot O_1 + \dots + O' \cdot O_n$
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	$(O_1 \otimes O_2) \cdot ((O'_1 \otimes O'_2) \cdot O_3) \triangleright ((O_1 \cdot O'_1) \otimes (O_2 \cdot O'_2)) \cdot O_3$

Table 12: Rules for sets.

Rule	Description
R-SET0	$\mathbf{U}(\sigma) \star \mathbf{U}(\tau) \triangleright \mathbf{U}(\sigma \times \tau)$

Table 13: Rules for sum operators.

Rule	Description
R-SUM-CONST0	$\sum_{x \in s} 0 \triangleright 0$
R-SUM-CONST1	$\sum_{x \in s} \mathbf{0}_{\mathcal{K}}(\sigma) \triangleright \mathbf{0}_{\mathcal{K}}(\sigma)$
R-SUM-CONST2	$\sum_{x \in s} \mathbf{0}_{\mathcal{B}}(\sigma) \triangleright \mathbf{0}_{\mathcal{B}}(\sigma)$
R-SUM-CONST3	$\sum_{x \in s} \mathbf{0}_{\mathcal{O}}(\sigma, \tau) \triangleright \mathbf{0}_{\mathcal{O}}(\sigma, \tau)$
R-SUM-CONST4	$\mathbf{1}_{\mathcal{O}}(\sigma) \triangleright \sum_{i \in \mathbf{U}(\sigma)} i\rangle \cdot \langle i $

Table 14: Rules for eliminating $\delta_{s,t}$.

Rule	Description
R-SUM-ELIM0	$i \text{ free in } t \vdash \sum_{i \in \mathbf{U}(\sigma)} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} \delta_{i,t} \triangleright \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} 1$
R-SUM-ELIM1	$i \text{ free in } t \vdash \sum_{i \in \mathbf{U}(\sigma)} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} (\delta_{i,t} \times a) \triangleright \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} a\{i/t\}$
R-SUM-ELIM2	$i \text{ free in } t \vdash \sum_{i \in \mathbf{U}(\sigma)} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} (\delta_{i,t} \cdot A) \triangleright \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} A\{i/t\}$
R-SUM-ELIM3	$i \text{ free in } t \vdash \sum_{i \in \mathbf{U}(\sigma)} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} (\delta_{i,t} \times a) \cdot A \triangleright \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} a\{i/t\} \cdot A\{i/t\}$
R-SUM-ELIM4	$\sum_{i \in M} \sum_{j \in M} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} \delta_{i,j} \triangleright \sum_{i \in M} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} 1$
R-SUM-ELIM5	$\sum_{i \in M} \sum_{j \in M} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} (\delta_{i,j} \times a) \triangleright \sum_{i \in M} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} a\{i/j\}$
R-SUM-ELIM6	$\sum_{i \in M} \sum_{j \in M} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} (\delta_{i,j} \cdot A) \triangleright \sum_{i \in M} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} A\{i/j\}$
R-SUM-ELIM7	$\sum_{i \in M} \sum_{j \in M} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} (\delta_{i,j} \times a) \cdot A \triangleright \sum_{i \in M} \sum_{k_1 \in s_1} \dots \sum_{k_n \in s_n} a\{i/j\} \cdot A\{i/j\}$

Table 15: Rules for pushing terms into sum operators

Rule	Description
R-SUM-PUSH0	$(\sum_{i \in M} a) \times b \triangleright \sum_{i \in M} (a \times b)$
R-SUM-PUSH1	$(\sum_{i \in M} a)^* \triangleright \sum_{i \in M} a^*$
R-SUM-PUSH2	$(\sum_{i \in M} X)^\dagger \triangleright \sum_{i \in M} X^\dagger$
R-SUM-PUSH3	$a.(\sum_{i \in M} X) \triangleright \sum_{i \in M} (a.X)$
R-SUM-PUSH4	$(\sum_{i \in M} a).X \triangleright \sum_{i \in M} (a.X)$
R-SUM-PUSH5	$(\sum_{i \in M} B) \cdot K \triangleright \sum_{i \in M} (B \cdot K)$
R-SUM-PUSH6	$(\sum_{i \in M} O) \cdot K \triangleright \sum_{i \in M} (O \cdot K)$
R-SUM-PUSH7	$(\sum_{i \in M} B) \cdot O \triangleright \sum_{i \in M} (B \cdot O)$
R-SUM-PUSH8	$(\sum_{i \in M} K) \cdot B \triangleright \sum_{i \in M} (K \cdot B)$
R-SUM-PUSH9	$(\sum_{i \in M} O_1) \cdot O_2 \triangleright \sum_{i \in M} (O_1 \cdot O_2)$
R-SUM-PUSH10	$B \cdot (\sum_{i \in M} K) \triangleright \sum_{i \in M} (B \cdot K)$
R-SUM-PUSH11	$O \cdot (\sum_{i \in M} K) \triangleright \sum_{i \in M} (O \cdot K)$
R-SUM-PUSH12	$B \cdot (\sum_{i \in M} O) \triangleright \sum_{i \in M} (B \cdot O)$
R-SUM-PUSH13	$K \cdot (\sum_{i \in M} B) \triangleright \sum_{i \in M} (K \cdot B)$
R-SUM-PUSH14	$O_1 \cdot (\sum_{i \in M} O_2) \triangleright \sum_{i \in M} (O_1 \cdot O_2)$
R-SUM-PUSH15	$(\sum_{i \in M} K_1) \otimes K_2 \triangleright \sum_{i \in M} (K_1 \otimes K_2)$
R-SUM-PUSH16	$(\sum_{i \in M} B_1) \otimes B_2 \triangleright \sum_{i \in M} (B_1 \otimes B_2)$
R-SUM-PUSH17	$(\sum_{i \in M} O_1) \otimes O_2 \triangleright \sum_{i \in M} (O_1 \otimes O_2)$
R-SUM-PUSH18	$K_1 \otimes (\sum_{i \in M} K_2) \triangleright \sum_{i \in M} (K_1 \otimes K_2)$
R-SUM-PUSH19	$B_1 \otimes (\sum_{i \in M} B_2) \triangleright \sum_{i \in M} (B_1 \otimes B_2)$
R-SUM-PUSH20	$O_1 \otimes (\sum_{i \in M} O_2) \triangleright \sum_{i \in M} (O_1 \otimes O_2)$

[YX] : Note: because we apply type checking on variables, and stick to unique bound variables, these pushing in operations are always sound.

Table 16: Rules for addition and index in sum

Rule	Description
R-SUM-ADD0	$\sum_{i \in M} (X_1 + \dots + X_n) \triangleright (\sum_{i \in M} X_1) + \dots + (\sum_{i \in M} X_n)$
R-SUM-ADD1	$Y_1 + \dots + \sum_{i \in M} (a.X) + \dots + \sum_{i \in M} X + Y_n \triangleright Y_1 + \dots + Y_n + \sum_{i \in M} (a+1).X$
R-SUM-ADD2	$Y_1 + \dots + \sum_{i \in M} X + \dots + \sum_{i \in M} (a.X) + Y_n \triangleright Y_1 + \dots + Y_n + \sum_{i \in M} (1+a).X$
R-SUM-ADD3	$Y_1 + \dots + \sum_{i \in M} (a.X) + \dots + \sum_{i \in M} (b.X) + Y_n \triangleright Y_1 + \dots + Y_n + \sum_{i \in M} (a+b).X$
R-SUM-INDEX0	$\sum_{i \in \mathbf{U}(\sigma \times \tau)} A \triangleright \sum_{j \in \mathbf{U}(\sigma)} \sum_{k \in \mathbf{U}(\tau)} A\{i/(j, k)\}$
R-SUM-INDEX1	$\sum_{i \in M_1 \star M_2} A \triangleright \sum_{j \in M_1} \sum_{k \in M_2} A\{i/(j, k)\}$

5 Diracoq language

```

cmd ::= Def(ID term)
      | Def(ID term type)
      | Var(ID term)
      | Check(term)
      | Show(ID)
      | ShowAll
      | Normalize(term)
      | CheckEq(term term)
type ::= Type | Arrow(type type)
       | Base
term ::= Type | fun(ID type term) | apply(term term) | ID

```

6 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 6.1 (permutation tree). *The syntax for permutation trees are inductively defined below:*

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

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 $\{i_1, \dots, i_n\}$ 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

```

apply(A, P) := match P with
  | E ⇒ A
  | [i1 : P1 ⋯ in : Pn] ⇒ A.head(apply(A.args[i1], Pi1) ⋯ apply(A.args[in], Pin))
end

```

7 Labelled Dirac Notation

Definition 7.1 (quantum registers).

$$R ::= x \mid (R, R) \mid \text{fst } R \mid \text{snd } R$$

We define the following relations for quantum registers:

- *R belongs to Q*, written as R in Q ,
- *R is disjoint with Q*, written as $R \parallel Q$.

Remark: We still have a speical algorithm deciding the relations.

Definition 7.2 (register set).

$$S ::= \emptyset \mid \{R\} \mid S \cup S \mid S \setminus S$$

Remark: $S_1 \cap S_2 \equiv S_1 \cup S_2 \setminus (S_1 \setminus S_2) \setminus (S_2 \setminus S_1)$

REG

$$\text{fst } (R_1, R_2) \triangleright R_1 \quad \text{snd } (R_1, R_2) \triangleright R_2 \quad (\text{fst } R, \text{snd } R) \triangleright R$$

RSET

$$\begin{aligned} S \cup \emptyset \triangleright S \quad S \cup S \triangleright S \quad \{\text{fst } R\} \cup \{\text{snd } R\} \triangleright R \\ S \setminus \emptyset \triangleright S \quad \emptyset \setminus S \triangleright \emptyset \quad S \setminus S \triangleright \emptyset \\ (S_1 \cup S_2) \setminus X \triangleright (S_1 \setminus X) \cup (S_2 \setminus X) \quad S_1 \setminus (S_2 \cup S_3) \triangleright (S_1 \setminus S_2) \setminus S_3 \end{aligned}$$

$$\frac{R_1 \text{ in } R_2}{\{R_1\} \cup \{R_2\} \triangleright \{R_2\}} \quad \frac{R_1 \text{ in } R_2}{\{R_1\} \setminus \{R_2\} \triangleright \emptyset}$$

$$\frac{R_1 \text{ in } R_2}{\{R_2\} \setminus \{R_1\} \triangleright (\{\text{fst } R_2\} \setminus \{R_1\}) \cup (\{\text{snd } R_2\} \setminus \{R_1\})} \quad \frac{R_1 \parallel R_2}{\{R_1\} \setminus \{R_2\} \triangleright \{R_1\}}$$

Definition 7.3 (labelled core language). *The **labelled core langauge** includes all symbols in the core language of Dirac notation, as well as symbols for the three new sorts.*

$$\begin{aligned} S &::= \tilde{B} \cdot \tilde{K} \\ \tilde{K}(\text{labelled ket}) &::= K_R \mid \tilde{B}^\dagger \mid S.\tilde{K} \mid \tilde{K} + \tilde{K} \mid \tilde{O} \cdot \tilde{K} \mid \tilde{K} \otimes \tilde{K} \\ \tilde{B}(\text{labelled bra}) &::= B_R \mid \tilde{K}^\dagger \mid S.\tilde{B} \mid \tilde{B} + \tilde{B} \mid \tilde{B} \cdot \tilde{O} \mid \tilde{B} \otimes \tilde{B} \\ \tilde{O}(\text{labelled operator}) &::= O_{R;R} \mid \tilde{K} \otimes \tilde{B} \mid \tilde{O}^\dagger \mid S.\tilde{O} \mid \tilde{O} + \tilde{O} \mid \tilde{O} \cdot \tilde{O} \mid \tilde{O} \otimes \tilde{O} \end{aligned}$$

In other words, we don't allow variables for labelled core language for now.

LABEL-CORE

We generally copied the symbols ($\dagger, S.\tilde{X}, +, \cdot, \otimes$) from the core language. Therefore we also need a copy of the corresponding rewriting rules.

TSR-DECOMP

$$\begin{aligned}
& |(s, t)\rangle_{(Q,R)} \triangleright |s\rangle_Q \otimes |t\rangle_R \quad \langle (s, t)|_{(Q,R)} \triangleright \langle s|_Q \otimes \langle t|_R \\
& \mathbf{0}_{\mathcal{K}(Q,R)} \triangleright \mathbf{0}_{\mathcal{K}Q} \otimes \mathbf{0}_{\mathcal{K}R} \quad \mathbf{0}_{\mathcal{B}(Q,R)} \triangleright \mathbf{0}_{\mathcal{B}Q} \otimes \mathbf{0}_{\mathcal{B}R} \\
& \mathbf{0}_{\mathcal{O}(Q,R);(S,T)} \triangleright \mathbf{0}_{\mathcal{O}(Q,S)} \otimes \mathbf{0}_{\mathcal{O}(R,T)} \\
& \mathbf{0}_{\mathcal{O}(Q_1,Q_2);R} \triangleright \mathbf{0}_{\mathcal{O}Q_1;\text{fst } R} \otimes \mathbf{0}_{\mathcal{O}Q_2;\text{fst } R} \quad \mathbf{0}_{\mathcal{O}Q;(R_1,R_2)} \triangleright \mathbf{0}_{\mathcal{O}\text{fst } Q;R_1} \otimes \mathbf{0}_{\mathcal{O}\text{snd } Q;R_2} \\
& \mathbf{1}_{\mathcal{O}(Q,R);(Q,R)} \triangleright \mathbf{1}_{\mathcal{O}Q;Q} \otimes \mathbf{1}_{\mathcal{O}R;R} \\
& \mathbf{1}_{\mathcal{O}(Q_1,Q_2);R} \triangleright \mathbf{1}_{\mathcal{O}Q_1;\text{fst } R} \otimes \mathbf{1}_{\mathcal{O}Q_2;\text{fst } R} \quad \mathbf{1}_{\mathcal{O}Q;(R_1,R_2)} \triangleright \mathbf{1}_{\mathcal{O}\text{fst } Q;R_1} \otimes \mathbf{1}_{\mathcal{O}\text{snd } Q;R_2} \\
& (K_1 \otimes K_2)_{(Q,R)} \triangleright K_{1Q} \otimes K_{2R} \quad (B_1 \otimes B_2)_{(Q,R)} \triangleright B_{1Q} \otimes B_{2R} \\
& (O_1 \otimes O_2)_{(Q,R);(S,T)} \triangleright O_{1Q;S} \otimes O_{2R;T} \\
& (O_1 \otimes O_2)_{(Q_1,Q_2);R} \triangleright O_{1Q_1;\text{fst } R} \otimes O_{2Q_2;\text{snd } R} \quad (O_1 \otimes O_2)_{Q;(R_1,R_2)} \triangleright O_{1\text{fst } Q;R_1} \otimes O_{2\text{snd } Q;R_2}
\end{aligned}$$

TSR-COMP

$$\begin{aligned}
& K_{1\text{fst } R} \otimes K_{1\text{snd } R} \triangleright (K_1 \otimes K_2)_R \quad B_{1\text{fst } R} \otimes B_{2\text{snd } R} \triangleright (B_1 \otimes B_2)_R \\
& O_{1\text{fst } Q;\text{fst } R} \otimes O_{2\text{snd } Q;\text{snd } R} \triangleright (O_1 \otimes O_2)_{Q;R}
\end{aligned}$$

DOT-TSR

$$\frac{R \parallel S}{O_{1Q;R} \cdot O_{2S;T} \triangleright O_{1Q;R} \otimes O_{2S;T}}$$

LABEL-LIFT

$$\begin{aligned}
& (K_R)^\dagger \triangleright (K^\dagger)_R \quad (B_R)^\dagger \triangleright (B^\dagger)_R \quad (O_{Q;R})^\dagger \triangleright (O^\dagger)_{Q;R} \\
& (K_R)^\top \triangleright (K^\top)_R \quad (B_R)^\top \triangleright (B^\top)_R \quad (O_{Q;R})^\top \triangleright (O^\top)_{Q;R} \\
& (S.K)_R \triangleright S.(K_R) \quad (S.B)_R \triangleright S.(B_R) \quad (S.O)_{Q;R} \triangleright S.(O_{Q;R}) \\
& (K_1 + K_2)_R \triangleright K_{1R} + K_{2R} \quad (B_1 + B_2)_R \triangleright B_{1R} + B_{2R} \quad (O_1 + O_2)_{Q;R} \triangleright O_{1Q;R} + O_{2Q;R} \\
& O_{1Q;R} \cdot O_{2R;S} \triangleright (O_1 \cdot O_2)_{Q;S} \quad O_{Q;R} \cdot K_R \triangleright (O \cdot K)_Q \quad B_Q \cdot O_{Q;R} \triangleright (B \cdot O)_R \\
& B_R \cdot K_R \triangleright B \cdot K \\
& (K \otimes B)_{Q;R} \triangleright K_Q \otimes B_R
\end{aligned}$$

OPT-EXT

I think the concept “cylinder extension” is only limited to endomorphisms. Besides, one quantum register should a sub-register of the other one, which is defined as follows:

Definition 7.4 (sub-register).

$$\text{fst } R \preceq R \quad \text{snd } R \preceq R \quad Q \preceq (Q, R) \quad R \preceq (Q, R) \quad \frac{Q \preceq R \quad R \preceq S}{Q \preceq S}$$

And we can further calculate the “position” of sub-register, which will be utilized during cylinder extension: assume Q is a sub-register of R , then the position of Q in R is a string defined as follows:

$$\begin{aligned}
& \text{pos}(\text{fst } R, R) = 0 \\
& \text{pos}(\text{snd } R, R) = 1 \\
& \text{pos}(Q, (Q, R)) = 0 \\
& \text{pos}(R, (Q, R)) = 1 \\
& \text{pos}(Q, S) = \text{pos}(R, S) \text{ pos}(Q, R)
\end{aligned}$$

Remark: For a well-formed quantum register, the sub-register position is well-defined.

Definition 7.5 (cylinder extension).

$$\text{ext}(O, \epsilon) \equiv O \qquad \text{ext}(O, p :: 0) \equiv O \otimes \mathbf{1}_O \qquad \text{ext}(O, p :: 1) \equiv \mathbf{1}_O \otimes O$$

CYLINDER-EXT

$$\frac{Q \text{ is a subterm of } R \text{ at } p}{O_{Q;Q} \cdot K_R \triangleright (\text{ext}(O, p) \cdot K)_R} \qquad \frac{Q \text{ is a subterm of } R \text{ at } p}{O_{1Q;Q} \cdot O_{2R;T} \triangleright (\text{ext}(O_1, p) \cdot O_2)_{R;T}}$$

$$\frac{R \text{ is a subterm of } Q \text{ at } p}{B_Q \cdot O_{R;R} \triangleright (B \cdot \text{ext}(O, p))_Q} \qquad \frac{R \text{ is a subterm of } Q \text{ at } p}{O_{1T;Q} \cdot O_{2R;T} \triangleright (O_1 \cdot \text{ext}(O_2, p))_{T;Q}}$$

7.1 Labelled Extended Language

Definition 7.6 (labelled extended language). *The **labelled extended language** consists of the symbols in labelled core language and unlabelled extended language, and add the new symbols of transpose and big-op for labelled bra, ket and operators, which is described in the following.*

$$\tilde{K} ::= \tilde{B}^\top \mid \sum_{i \in M} \tilde{K} \qquad \tilde{B} ::= \tilde{K}^\top \mid \sum_{i \in M} \tilde{B} \qquad \tilde{O} ::= \tilde{O}^\top \mid \sum_{i \in M} \tilde{O}$$

LABEL-SUM

$$\left(\sum_{i \in M} K \right)_R \triangleright \sum_{i \in M} (K_R) \qquad \left(\sum_{i \in M} B \right)_R \triangleright \sum_{i \in M} (B_R) \qquad \left(\sum_{i \in M} O \right)_{Q;R} \triangleright \sum_{i \in M} (O_{Q;R})$$

LABEL-TEMP

[YX] : These are ad-hoc rules for the examples for now. They are still not organized and require further investigations.

$$\frac{R \text{ in } Q}{O_{P;R} \cdot (K_Q \otimes \tilde{K}') \triangleright (O_{P;R} \cdot K_Q) \otimes \tilde{K}'} \qquad \frac{R \text{ in } Q}{O_{P;R} \cdot (K_Q \otimes \tilde{B}) \triangleright (O_{P;R} \cdot K_Q) \otimes \tilde{B}}$$

$$\frac{R \text{ in } Q}{O_{1P;R} \cdot (O_{2Q;T} \otimes \tilde{O}_3) \triangleright (O_{1P;R} \cdot O_{2Q;T}) \otimes \tilde{O}_3}$$

$$\frac{Q \text{ in } R}{(B_R \otimes \tilde{B}') \cdot O_{Q;P} \triangleright (B_R \cdot O_{Q;P}) \otimes \tilde{B}'} \qquad \frac{Q \text{ in } R}{(O_{1T;R} \otimes \tilde{O}') \cdot O_{2Q;P} \triangleright (O_{1T;R} \cdot O_{2Q;P}) \otimes \tilde{O}'}$$

$$B_R \cdot (K_R \otimes \tilde{B}') \triangleright (B_R \cdot K_R) \cdot \tilde{B}' \qquad (\tilde{K}' \otimes B_R) \cdot K_R \triangleright (B_R \cdot K_R) \cdot \tilde{K}'$$

References

- [1] Walliam Shakespeard. Halmet. 2666.