DiracDec C++/Coq Implementation[1]

January 3, 2025

1 Consideration

- We can emphasize that we are moving from pure term rewriting to a hybrid algorithm.
- AC equivalence and alpha equivalence.
- Efficient AC equivalence checking by sorting on syntax.
- Sorting requires that alpha equivalent terms have the same syntax.
- de Bruijn expression satisfies this requirement, but the typing and substitution becomes very complicated in the type index scenario.
- only compute de Bruijn form in the equivalence checking and sorting phase.
- pipeline:
  1. preprocessing
  2. rename unique bound variable names
  3. 1st rewritings
  4. variable expansion
  5. 2nd rewritings
  6. sorting modulo bound variables
  7. sorting successive sum. The order depends on there occurances in the last sorting result. If no occurance, then the order will depend on the set (for sum) and the type (for lambda abstraction only).
  8. computing de Bruijn
- I found that typing of inner bound variables also requires modifying the context. Need to modify the tree visiting algorithm (Typed term rewriting with bound variables is not that easy)

2 TODO

- implemented trace output for the whole pipeline.
- better output.
- add Mathematica support for scalars
- output trace to Coq
- consider labelled Dirac notations
- better trace output
- better error logic
- better format output
- remove term bank and hash consing

3 Things to Note

- The context should also be maintained during rewriting matching.
- We don't allow eta reduction. It will intertwine with SUM-SWAP and break the confluence.
- In each Delta reduction, the bound variables are replaced with unique variables. This should help solve the problem of conflicting variable names during substitution.
- I guess it may still be necessary to try all different bound variable assignments.

4 Language Syntax

Definition 4.1 (syntax)**.** *The syntax for type indices are defined as*

$$\sigma ::= x \mid \sigma_1 \times \sigma_2 \mid \mathsf{Qdit}_n.$$

The syntax for Dirac notation types is defined as

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

The syntax for Dirac notation terms is defined as

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

Here i is a natural number and $\$i$ represents the i-th bound variable in de Bruijn notation. 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 $\sum_s X$ to denote $\sum_{x \in s} \lambda x : T.X$ as well.

5 Typing System

The type checking of Dirac notations involves maintaining a well-formed environment and context $E[\Gamma]$, which specifies the definitions and typing assumtpions for variables. The environment and the conetxt are defined as follows.

Definition 5.1 (environment and context)**.**

$$\begin{aligned} E &::= [\,] \mid E; x : \mathsf{Index} \mid E; T : \mathsf{Type} \mid E; x : T \mid E; x := t : T. \\ \Gamma &::= [\,] \mid \Gamma; x : \mathsf{Index} \mid \Gamma; x : T. \end{aligned}$$

Note that in the following rules, $x : \mathsf{Index}$ and $x : \mathsf{Type}$ are not considered as $x : T$ judgements.

$$\textbf{W-Empty} \qquad \frac{}{\mathcal{WF}([])[]}$$

$$\textbf{W-AssumE-Index} \quad \frac{\mathcal{WF}(E)[] \qquad x \notin E}{\mathcal{WF}(E; x : \mathsf{Index})[]} \qquad\qquad \textbf{W-AssumE-Type} \quad \frac{\mathcal{WF}(E)[] \qquad x \notin E}{\mathcal{WF}(E; x : \mathsf{Type})[]}$$

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

$$\textbf{W-AssumC-Index} \quad \frac{\mathcal{WF}(E)[\Gamma]}{\mathcal{WF}(E)[\Gamma; x : \mathsf{Index}]} \qquad\qquad \textbf{W-AssumC-Term} \quad \frac{E[\Gamma] \vdash T : \mathsf{Type}}{\mathcal{WF}(E)[\Gamma; x : T]}$$

Figure 1: Rules for a well-formed environment and context.

$$\textbf{Index-Var} \quad \frac{\mathcal{WF}(E)[\Gamma] \qquad x : \mathsf{Index} \in E[\Gamma]}{E[\Gamma] \vdash x : \mathsf{Index}} \qquad\qquad \textbf{Index-Prod} \quad \frac{E[\Gamma] \vdash \sigma : \mathsf{Index} \qquad E[\Gamma] \vdash \tau : \mathsf{Index}}{E[\Gamma] \vdash \sigma \times \tau : \mathsf{Index}}$$

$$\textbf{Index-Qudit} \quad \frac{\mathcal{WF}(E)[\Gamma]}{E[\Gamma] \vdash \mathsf{Qudit}_n : \mathsf{Index}}$$

Figure 2: Rules for type index.

$$\textbf{Type-Arrow} \quad \frac{E[\Gamma] \vdash T : \mathsf{Type} \qquad E[\Gamma] \vdash U : \mathsf{Type}}{E[\Gamma] \vdash T \to U : \mathsf{Type}} \qquad\qquad \textbf{Type-Index} \quad \frac{E[\Gamma; x : \mathsf{Index}] \vdash U : \mathsf{Type}}{E[\Gamma] \vdash \forall x.U : \mathsf{Type}}$$

$$\textbf{Type-Basis} \quad \frac{E[\Gamma] \vdash \sigma : \mathsf{Index}}{E[\Gamma] \vdash \mathsf{Basis}(\sigma) : \mathsf{Type}}$$

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

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

$$\textbf{Type-Set} \quad \frac{E[\Gamma] \vdash \sigma : \mathsf{Index}}{E[\Gamma] \vdash \mathsf{Set}(\sigma) : \mathsf{Type}}$$

Figure 3: Rules for types.

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

$$\textbf{Lam}\qquad \frac{E[\Gamma; x : T] \vdash t : U}{E[\Gamma] \vdash (\lambda x : T.t) : T \to U} \qquad\qquad \textbf{Index}\qquad \frac{E[\Gamma; x : \mathsf{Index}] \vdash t : U}{E[\Gamma] \vdash (\mu x.t) : \forall x.U}$$

$$\textbf{App-Arrow}\qquad \frac{E[\Gamma] \vdash t : U \to T \qquad E[\Gamma] \vdash u : U}{E[\Gamma] \vdash (t\ u) : T} \qquad\qquad \textbf{App-Index}\qquad \frac{E[\Gamma] \vdash t : \forall x.U \qquad E[\Gamma] \vdash u : \mathsf{Index}}{E[\Gamma] \vdash (t\ u) : U\{x/u\}}$$

Figure 4: Rules for variable and function typings. Here $U[u]$ means instantiate the first bound variable with u in U.

$$\textbf{Pair-Base}\qquad \frac{E[\Gamma] \vdash s : \mathsf{Basis}(\sigma) \qquad E[\Gamma] \vdash t : \mathsf{Basis}(\tau)}{E[\Gamma] \vdash (s, t) : \mathsf{Basis}(\sigma \times \tau)}$$

Figure 5: Typing rules for Basis.

$$\textbf{Sca-0}\qquad \frac{\mathcal{WF}(E)[\Gamma]}{E[\Gamma] \vdash 0 : \mathcal{S}} \qquad\qquad \textbf{Sca-1}\qquad \frac{\mathcal{WF}(E)[\Gamma]}{E[\Gamma] \vdash 1 : \mathcal{S}}$$

$$\textbf{Sca-Delta}\qquad \frac{E[\Gamma] \vdash s : \mathsf{Basis}(\sigma) \qquad E[\Gamma] \vdash t : \mathsf{Basis}(\sigma)}{E[\Gamma] \vdash \delta_{s,t} : \mathcal{S}}$$

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

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

Figure 6: Scalar typing rules.

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

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

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

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

Figure 7: Ket typing rules.

Bra-0
$$\frac{E[\Gamma] \vdash \sigma : \mathsf{Index}}{E[\Gamma] \vdash \mathbf{0}_{\mathcal{B}}(\sigma) : \mathcal{B}(\sigma)}$$

Bra-Base
$$\frac{E[\Gamma] \vdash t : \mathsf{Basis}(\sigma)}{E[\Gamma] \vdash \langle t| : \mathcal{B}(\sigma)}$$

Bra-Adj
$$\frac{E[\Gamma] \vdash K : \mathcal{K}(\sigma)}{E[\Gamma] \vdash K^{\dagger} : \mathcal{B}(\sigma)}$$

Bra-Scr
$$\frac{E[\Gamma] \vdash a : \mathcal{S} \qquad E[\Gamma] \vdash B : \mathcal{B}(\sigma)}{E[\Gamma] \vdash a.B : \mathcal{B}(\sigma)}$$

Bra-Add
$$\frac{E[\Gamma] \vdash B_i : \mathcal{B}(\sigma) \text{ for all } i}{E[\Gamma] \vdash B_1 + \cdots + B_n : \mathcal{B}(\sigma)}$$

Bra-MulB
$$\frac{E[\Gamma] \vdash B : \mathcal{K}(\sigma) \qquad E[\Gamma] \vdash O : \mathcal{O}(\sigma, \tau)}{E[\Gamma] \vdash B \cdot O : \mathcal{B}(\tau)}$$

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

Figure 8: Bra typing rules.

Opt-0
$$\frac{E[\Gamma] \vdash \sigma : \mathsf{Index} \qquad E[\Gamma] \vdash \tau : \mathsf{Index}}{E[\Gamma] \vdash \mathbf{0}_{\mathcal{O}}(\sigma, \tau) : \mathcal{O}(\sigma, \tau)}$$

Opt-1
$$\frac{E[\Gamma] \vdash \sigma : \mathsf{Index}}{E[\Gamma] \vdash \mathbf{1}_{\mathcal{O}}(\sigma) : \mathcal{O}(\sigma, \sigma)}$$

Opt-Adj
$$\frac{E[\Gamma] \vdash O : \mathcal{O}(\sigma, \tau)}{E[\Gamma] \vdash O^{\dagger} : \mathcal{O}(\tau, \sigma)}$$

Opt-Scr
$$\frac{E[\Gamma] \vdash a : \mathcal{S} \qquad E[\Gamma] \vdash O : \mathcal{O}(\sigma, \tau)}{E[\Gamma] \vdash a.O : \mathcal{O}(\sigma, \tau)}$$

Opt-Add
$$\frac{E[\Gamma] \vdash O_i : \mathcal{O}(\sigma, \tau) \text{ for all } i}{E[\Gamma] \vdash O_1 + \cdots + O_n : \mathcal{O}(\sigma, \tau)}$$

Opt-Outer
$$\frac{E[\Gamma] \vdash K : \mathcal{K}(\sigma) \qquad E[\Gamma] \vdash B : \mathcal{B}(\tau)}{E[\Gamma] \vdash K \cdot B : \mathcal{O}(\sigma, \tau)}$$

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

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

Figure 9: Operator typing rules.

Set-U
$$\frac{E[\Gamma] \vdash \sigma : \mathsf{Index}}{E[\Gamma] \vdash \mathbf{U}(\sigma) : \mathsf{Set}(\sigma)}$$

Set-Prod
$$\frac{E[\Gamma] \vdash A : \mathsf{Set}(\sigma) \qquad E[\Gamma] \vdash B : \mathsf{Set}(\tau)}{E[\Gamma] \vdash A \star B : \mathsf{Set}(\sigma \times \tau)}$$

Figure 10: Set typing rules.

$$\textbf{Sum-Scalar}\qquad \frac{E[\Gamma] \vdash s : \mathsf{Set}(\sigma) \qquad E[\Gamma] \vdash f : \mathsf{Basis}(\sigma) \to \mathcal{S}}{E[\Gamma] \vdash \sum_s f : \mathcal{S}}$$

$$\textbf{Sum-Ket}\qquad \frac{E[\Gamma] \vdash s : \mathsf{Set}(\sigma) \qquad E[\Gamma] \vdash f : \mathsf{Basis}(\sigma) \to \mathcal{K}(\tau)}{E[\Gamma] \vdash \sum_s f : \mathcal{K}(\tau)}$$

$$\textbf{Sum-Bra}\qquad \frac{E[\Gamma] \vdash s : \mathsf{Set}(\sigma) \qquad E[\Gamma] \vdash f : \mathsf{Basis}(\sigma) \to \mathcal{B}(\tau)}{E[\Gamma] \vdash \sum_s f : \mathcal{B}(\tau)}$$

$$\textbf{Sum-Opt}\qquad \frac{E[\Gamma] \vdash s : \mathsf{Set}(\sigma) \qquad E[\Gamma] \vdash f : \mathsf{Basis}(\sigma) \to \mathcal{O}(\tau, \rho)}{E[\Gamma] \vdash \sum_s f : \mathcal{O}(\tau, \rho)}$$

Figure 11: Sum typing rules.

6 Conversions and Reductions

$$\frac{E[\Gamma] \vdash K : \mathcal{K}(\sigma)}{E[\Gamma] \vdash K \ \triangleright\ \sum_{i \in \mathbf{U}(\sigma)} (\langle i| \cdot K).\, |i\rangle}$$

$$\frac{E[\Gamma] \vdash B : \mathcal{B}(\sigma)}{E[\Gamma] \vdash B \ \triangleright\ \sum_{i \in \mathbf{U}(\sigma)} (B \cdot |i\rangle).\, \langle i|}$$

$$\frac{E[\Gamma] \vdash O : \mathcal{O}(\sigma, \tau)}{E[\Gamma] \vdash O \ \triangleright\ \sum_{i \in \mathbf{U}(\sigma)} \sum_{j \in \mathbf{U}(\tau)} (\langle i| \cdot O \cdot |j\rangle).(|i\rangle \cdot \langle j|)}$$

Beta-Arrow $$\frac{}{E[\Gamma] \vdash ((\lambda x : T.t)\ u) \ \triangleright\ t\{x/u\}}$$ **Beta-Index** $$\frac{}{E[\Gamma] \vdash ((\mu x.t)\ u) \ \triangleright\ t\{x/u\}}$$

Delta $$\frac{\mathcal{WF}(E)[\Gamma] \qquad (c := t : T) \in E}{E[\Gamma] \vdash c \ \triangleright\ t}$$

Figure 12: Conversions and reductions for the typed lambda calculus.

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	$\delta_{s,t}^* \triangleright \delta_{s,t}$
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	$\langle s \vert \cdot \vert t \rangle \triangleright \delta_{s,t}$
R-DOT7	$(B_1 \otimes B_2) \cdot \vert (s,t) \rangle \triangleright (B_1 \cdot \vert s \rangle) \times (B_2 \cdot \vert t \rangle)$
R-DOT8	$\langle (s,t) \vert \cdot (K_1 \otimes K_2) \triangleright (\langle s \vert \cdot K_1) \times (\langle t \vert \cdot K_2)$
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	$\langle (s,t) \vert \cdot ((O_1 \otimes O_2) \cdot K) \triangleright (((\langle s \vert \cdot O_1) \otimes (\langle t \vert \cdot O_2)) \cdot K$
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	$Y_1 + \cdots + X + \cdots + X + \cdots + Y_n \triangleright Y_1 + \cdots + Y_n + \cdots + (1+1).X$

Continued on the next page

Rule	Description
	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).
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	$\langle t\vert^\dagger \ \triangleright\ \vert t\rangle$
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	$\vert t\rangle^\dagger \ \triangleright\ \langle t\vert$
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	$\vert s\rangle \otimes \vert t\rangle \ \triangleright\ \vert(s, t)\rangle$
R-TSRB0	$B : \mathcal{B}(\tau) \Rightarrow \mathbf{0}_{\mathcal{B}}(\sigma) \otimes B \ \triangleright\ \mathbf{0}_{\mathcal{B}}(\sigma \times \tau)$

Continued on the next page

Rule	Description
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	$(O_1 \otimes O_2) \cdot \|(s,t)\rangle \ \triangleright\ (O_1 \cdot \|s\rangle) \otimes (O_2 \cdot \|t\rangle)$
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	$(B_1 + \cdots + B_n) \cdot O \ \triangleright\ B_1 \cdot O + \cdots + B_n \cdot O$
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	$\langle (s,t)\| \cdot (O_1 \otimes O_2) \ \triangleright\ (\langle s\| \cdot O_1) \otimes (\langle t\| \cdot O_2)$
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	$K : \mathcal{K}(\sigma) \Rightarrow K \cdot \mathbf{0}_{\mathcal{B}}(\tau) \ \triangleright\ \mathbf{0}_{\mathcal{O}}(\sigma,\tau)$
	Continued on the next page

Rule	Description
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	$(O_1 \otimes O_2) \cdot ((O'_1 \otimes O'_2) \cdot O_3) \ \triangleright\ ((O_1 \cdot O'_1) \otimes (O_2 \cdot O'_2)) \cdot O_3$

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	$\mathbf{1}_{\mathcal{O}}(\sigma) \ \triangleright\ \sum_{i \in \mathbf{U}(\sigma)} \vert i \rangle \cdot \langle i \vert$

Table 14: Rules for eliminating $\delta_{s,t}$. Note that these rules will match the δ operator modulo the commutativity of its arguments.

Rule	Description
R-SUM-ELIM0	i free in $t \vdash \sum_{i \in \mathbf{U}(\sigma)} \sum_{k_1 \in s_1} \cdots \sum_{k_n \in s_n} \delta_{i,t} \ \triangleright\ \sum_{k_1 \in s_1} \cdots \sum_{k_n \in s_n} 1$
R-SUM-ELIM1	i free in $t \vdash \sum_{i \in \mathbf{U}(\sigma)} \sum_{k_1 \in s_1} \cdots \sum_{k_n \in s_n} (a_1 \times \cdots \times \delta_{i,t} \times \cdots \times a_n)$
	$\triangleright\ \sum_{k_1 \in s_1} \cdots \sum_{k_n \in s_n} a_1\{i/t\} \times \cdots \times a_n\{i/t\}$

Continued on the next page

Rule	Description
R-SUM-ELIM2	i free in $t \vdash \sum_{i\in\mathbf{U}(\sigma)} \sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}(\delta_{i,t}.A) \triangleright \sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n} A\{i/t\}$
R-SUM-ELIM3	i free in $t \vdash \sum_{i\in\mathbf{U}(\sigma)} \sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}(a_1\times\cdots\times\delta_{i,t}\times\cdots\times a_n).A$ $\triangleright \sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}(a_1\{i/t\}\times\cdots\times a_n\{i/t\}).A\{i/t\}$
R-SUM-ELIM4	$\sum_{i\in M}\sum_{j\in M}\sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}\delta_{i,j} \triangleright \sum_{j\in M}\sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n} 1$
R-SUM-ELIM5	$\sum_{i\in M}\sum_{j\in M}\sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}(a_1\times\cdots\times\delta_{i,j}\times\cdots\times a_n)$ $\triangleright \sum_{j\in M}\sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}(a_1\{j/i\}\times\cdots\times a_n\{j/i\})$
R-SUM-ELIM6	$\sum_{i\in M}\sum_{j\in M}\sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}(\delta_{i,j}.A) \triangleright \sum_{j\in M}\sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n} A\{j/i\}$
R-SUM-ELIM7	$\sum_{i\in M}\sum_{j\in M}\sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}(a_1\times\cdots\times\delta_{i,j}\times\cdots\times a_n).A$ $\triangleright \sum_{j\in M}\sum_{k_1\in s_1}\cdots\sum_{k_n\in s_n}(a_1\{j/i\}\times\cdots\times a_n\{j/i\}).A\{j/i\}$

Table 15: Rules for pushing terms into sum operators

Rule	Description
R-SUM-PUSH0	$b_1\times\cdots\times(\sum_{i\in M} a)\times\cdots\times b_n \triangleright \sum_{i\in M}(b_1\times\cdots\times a\times\cdots\times b_n)$
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	$(\sum_{i\in M} X_1)\otimes X_2 \triangleright \sum_{i\in M}(X_1\otimes X_2)$
R-SUM-PUSH16	$X_1\otimes(\sum_{i\in M} X_2) \triangleright \sum_{i\in M}(X_1\otimes X_2)$

[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-ADDS0	$\sum_{i\in M}(a_1+\cdots+a_n) \triangleright (\sum_{i\in M} a_1)+\cdots+(\sum_{i\in M} a_n)$
R-SUM-ADD0	$\sum_{i\in M}(X_1+\cdots+X_n) \triangleright (\sum_{i\in M} X_1)+\cdots+(\sum_{i\in M} X_n)$
R-SUM-ADD1	$Y_1+\cdots+\sum_{i\in M}(a.X)+\cdots+\sum_{i\in M} X+Y_n \triangleright Y_1+\cdots+Y_n+\sum_{i\in M}(a+1).X$
R-SUM-ADD2	$Y_1+\cdots+\sum_{i\in M} X+\cdots+\sum_{i\in M}(a.X)+Y_n \triangleright Y_1+\cdots+Y_n+\sum_{i\in M}(1+a).X$
R-SUM-ADD3	$Y_1+\cdots+\sum_{i\in M}(a.X)+\cdots+\sum_{i\in M}(b.X)+Y_n \triangleright Y_1+\cdots+Y_n+\sum_{i\in M}(a+b).X$
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	$\sum_{i\in M_1\star M_2} A \triangleright \sum_{j\in M_1}\sum_{k\in M_2} A\{i/(j,k)\}$

[YX] : The rules R-SUM-ADD1 to R-SUM-ADD3 needs identical sum terms, which requires the rewriting to be after the alpha normalization. We don't implement them for now.

$$\frac{E[\Gamma] \vdash B : \mathcal{B}(\sigma) \qquad E[\Gamma] \vdash K : \mathcal{K}(\sigma)}{E[\Gamma] \vdash B \circ K : \mathcal{S}}$$

Figure 13: The typing rules for preprocessing

7 Preprocessing

8 `Diracoq` language

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

Comment `(* ... *)` can be inserted between commands.
These are tye parsing rules for different expressions:

- `Def ID := term.` — `Def(ID term)`
- `Def ID := term : type.` — `Def(ID term type)`
- `Var ID : type.` — `Var(ID type)`
- `Check ID.` — `Check(term)`
- `Show ID.` — `Show(ID)`
- `ShowAll.` — `ShowAll`
- `Normalize term.` — `Normalize(term)`
- `Normalize term with trace.` — `Normalize(term Trace)`
- `Check term = term.` — `CheckEq(term term)`
- `T1 -> T2` — `Arrow(T1 T2).`
- `forall x. T` — `Forall(x T)`
- `(e1, e2)` — `PAIR(e1 e2)`
- `fun x : T => e` — `fun(x T e)`
- `idx x => e` — `idx(x e)`
- `e1 @ e2` — `COMPO(e1 e2)`
- `e1 + ... + en` — `ADDG(e1 ... en)`
- `e1 * ... * en` — `STAR(e1 ... en)`
- `e1^*` — `CONJ(e1)`
- `delta(e1, e2)` — `DELTA(e1 e2)`
- `|e>` — `KET(e)`
- `<e|` — `BRA(e)`
- `e1^D` — `ADJ(e1)`

- `e1.e2` — `SCR(e1 e2)`
- `Sum(i in s, e)` — `SSUM(i s e)`

These symbols will be transformed into internal language:

- $A \circ B$: $S \circ S$, $S \circ K$, $S \circ B$, $S \circ O$, $K \circ S$, $K \circ K$, $K \circ B$, $B \circ S$, $B \circ K$, $B \circ B$, $B \circ O$, $O \circ S$, $O \circ K$, $O \circ O$, $f \circ a$ (arrow), $f \circ a$ (index).
- `STAR(a ... b)` : $\sigma_1 \times \sigma_2$, $\mathsf{MULS}(a \cdots b)$, $O_1 \otimes O_2$, $M_1 \star M_2$
- `ADDG(e1 ... en)` : $\mathsf{ADDS}(e1 \cdots en)$, $\mathsf{ADD}(e1 \cdots en)$
- `SSUM(i S e)` : `SUM(s FUN(i T e))`

9 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 9.1 (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

```
apply($A, P$) := match $P$ with
                 | $\mathsf{E} \Rightarrow A$
                 | $[i_1 : P_1\ \cdots\ i_n : P_n] \Rightarrow$ A.head(apply(A.args[$i_1$], $P_{i_1}$) $\cdots$ apply(A.args[$i_n$], $P_{i_n}$))
                 end
```

10 Labelled Dirac Notation

Definition 10.1 (quantum registers)**.**

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

We define the following relations for quantum registers:

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

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

Definition 10.2 (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

$$\mathsf{fst}\ (R_1, R_2) \triangleright R_1 \qquad \mathsf{snd}\ (R_1, R_2) \triangleright R_2 \qquad (\mathsf{fst}\ R, \mathsf{snd}\ R) \triangleright R$$

RSET

$$\begin{gathered} S \cup \emptyset \triangleright S \qquad S \cup S \triangleright S \qquad \{\mathsf{fst}\ R\} \cup \{\mathsf{snd}\ R\} \triangleright R \\ S \setminus \emptyset \triangleright S \qquad \emptyset \setminus S \triangleright \emptyset \qquad S \setminus S \triangleright \emptyset \\ (S_1 \cup S_2) \setminus X \triangleright (S_1 \setminus X) \cup (S_2 \setminus X) \qquad S_1 \setminus (S_2 \cup S_3) \triangleright (S_1 \setminus S_2) \setminus S_3 \end{gathered}$$

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

$$\frac{R_1 \text{ in } R_2}{\{R_2\} \setminus \{R_1\} \triangleright (\{\mathsf{fst}\ R_2\} \setminus \{R_1\}) \cup (\{\mathsf{snd}\ R_2\} \setminus \{R_1\})} \qquad \frac{R_1 \| R_2}{\{R_1\} \setminus \{R_2\} \triangleright \{R_1\}}$$

Definition 10.3 (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.*

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

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

$$|(s,t)\rangle_{(Q,R)} \triangleright |s\rangle_Q \otimes |t\rangle_R \qquad \langle(s,t)|_{(Q,R)} \triangleright \langle s|_Q \otimes \langle s|_R$$
$$\mathbf{0}_{\mathcal{K}(Q,R)} \triangleright \mathbf{0}_{\mathcal{K}Q} \otimes \mathbf{0}_{\mathcal{K}R} \qquad \mathbf{0}_{\mathcal{B}(Q,R)} \triangleright \mathbf{0}_{\mathcal{B}Q} \otimes \mathbf{0}_{\mathcal{B}R}$$
$$\mathbf{0}_{\mathcal{O}(Q,R);(S,T)} \triangleright \mathbf{0}_{\mathcal{O}(Q,S)} \otimes \mathbf{0}_{\mathcal{O}(R,T)}$$
$$\mathbf{0}_{\mathcal{O}(Q_1,Q_2);R} \triangleright \mathbf{0}_{\mathcal{O}Q_1;\mathsf{fst}\ R} \otimes \mathbf{0}_{\mathcal{O}Q_2;\mathsf{fst}\ R} \qquad \mathbf{0}_{\mathcal{O}Q;(R_1,R_2)} \triangleright \mathbf{0}_{\mathcal{O}\mathsf{fst}\ Q;R_1} \otimes \mathbf{0}_{\mathcal{O}\mathsf{snd}\ Q;R_2}$$
$$\mathbf{1}_{\mathcal{O}(Q,R);(Q,R)} \triangleright \mathbf{1}_{\mathcal{O}Q;Q} \otimes \mathbf{1}_{\mathcal{O}R;R}$$
$$\mathbf{1}_{\mathcal{O}(Q_1,Q_2);R} \triangleright \mathbf{1}_{\mathcal{O}Q_1;\mathsf{fst}\ R} \otimes \mathbf{1}_{\mathcal{O}Q_2;\mathsf{fst}\ R} \qquad \mathbf{1}_{\mathcal{O}Q;(R_1,R_2)} \triangleright \mathbf{1}_{\mathcal{O}\mathsf{fst}\ Q;R_1} \otimes \mathbf{1}_{\mathcal{O}\mathsf{snd}\ Q;R_2}$$

$$(K_1 \otimes K_2)_{(Q,R)} \triangleright K_{1Q} \otimes K_{2R} \qquad (B_1 \otimes B_2)_{(Q,R)} \triangleright B_{1Q} \otimes B_{2R}$$
$$(O_1 \otimes O_2)_{(Q,R);(S,T)} \triangleright O_{1Q;S} \otimes O_{2R;T}$$
$$(O_1 \otimes O_2)_{(Q_1,Q_2);R} \triangleright O_{1Q_1;\mathsf{fst}\ R} \otimes O_{2Q_2;\mathsf{snd}\ R} \qquad (O_1 \otimes O_2)_{Q;(R_1,R_2)} \triangleright O_{1\mathsf{fst}\ Q;R_1} \otimes O_{2\mathsf{snd}\ Q;R_2}$$

TSR-COMP

$$K_{1\mathsf{fst}\ R} \otimes K_{1\mathsf{snd}\ R} \triangleright (K_1 \otimes K_2)_R \qquad B_{1\mathsf{fst}\ R} \otimes B_{2\mathsf{snd}\ R} \triangleright (B_1 \otimes B_2)_R$$
$$O_{1\mathsf{fst}\ Q;\mathsf{fst}\ R} \otimes O_{2\mathsf{snd}\ Q;\mathsf{snd}\ R} \triangleright (O_1 \otimes O_2)_{Q;R}$$

DOT-TSR

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

LABEL-LIFT

$$(K_R)^\dagger \triangleright (K^\dagger)_R \qquad (B_R)^\dagger \triangleright (B^\dagger)_R \qquad (O_{Q;R})^\dagger \triangleright (O^\dagger)_{Q;R}$$
$$(K_R)^\top \triangleright (K^\top)_R \qquad (B_R)^\top \triangleright (B^\top)_R \qquad (O_{Q;R})^\top \triangleright (O^\top)_{Q;R}$$
$$(S.K)_R \triangleright S.(K_R) \qquad (S.B)_R \triangleright S.(B_R) \qquad (S.O)_{Q;R} \triangleright S.(O_{Q;R})$$
$$(K_1 + K_2)_R \triangleright K_{1R} + K_{2R} \qquad (B_1 + B_2)_R \triangleright B_{1R} + B_{2R} \qquad (O_1 + O_2)_{Q;R} \triangleright O_{1Q;R} + O_{2Q;R}$$

$$O_{1Q;R} \cdot O_{2R;S} \triangleright (O_1 \cdot O_2)_{Q;S} \qquad O_{Q;R} \cdot K_R \triangleright (O \cdot K)_Q \qquad B_Q \cdot O_{Q;R} \triangleright (B \cdot O)_R$$
$$B_R \cdot K_R \triangleright B \cdot K$$
$$(K \otimes B)_{Q;R} \triangleright K_Q \otimes B_R$$

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 10.4 (sub-register)**.**

$$\mathsf{fst}\ R \preceq R \qquad \mathsf{snd}\ R \preceq R \qquad Q \preceq (Q,R) \qquad R \preceq (Q,R) \qquad \frac{Q \preceq R \qquad R \preceq S}{Q \preceq S}$$

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:

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

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

Definition 10.5 (cylinder extension)**.**

$$\mathit{ext}(O, \epsilon) \equiv O \qquad \mathit{ext}(O, p :: 0) \equiv O \otimes \mathbf{1}_{\mathcal{O}} \qquad \mathit{ext}(O, p :: 1) \equiv \mathbf{1}_{\mathcal{O}} \otimes O$$

CYLINDER-EXT

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

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

10.1 Labelled Extended Language

Definition 10.6 (labelled extended language)**.** *The* ***labelled extended language*** *consists of the symbols in labelled core langauge 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

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

LABEL-TEMP

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

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

$$\frac{R \textsf{ in } Q}{{O_1}_{P;R} \cdot ({O_2}_{Q;T} \otimes \tilde{O}_3) \triangleright ({O_1}_{P;R} \cdot {O_2}_{Q;T}) \otimes \tilde{O}_3}$$

$$\frac{Q \textsf{ in } R}{(B_R \otimes \tilde{B}') \cdot O_{Q;P} \triangleright (B_R \cdot O_{Q;P}) \otimes \tilde{B}'} \qquad \frac{Q \textsf{ in } R}{({O_1}_{T;R} \otimes \tilde{O}') \cdot {O_2}_{Q;P} \triangleright ({O_1}_{T;R} \cdot {O_2}_{Q;P}) \otimes \tilde{O}'}$$

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

References

[1] Walliam Shakespeard. Halmet. 2666.