

# Chapter 1

## 20240315

### 1.1 Motivation

[YX] : One important idea: mentioning that registers are just like variables in classical programming language.

### 1.2 Core Language: DN

The core language DN consists of limited symbols for Dirac notation, and its term rewriting system is expressed in standard AC-rewriting with congruence proved.

**Definition 1.2.1** (atomic basis signature). *The **atomic basis signature**  $\Sigma_B$  is an arbitrary signature.*

**Definition 1.2.2** (complex scalar signature). *The **complex scalar signature**  $\Sigma_S$  contains constant symbols  $0, 1, \frac{1}{2}$ , a unary symbol  $*$  and binary symbols  $+$ ,  $\times$ .*

**Definition 1.2.3** (language of Dirac Notation). *The **language of Dirac Notation**, denoted as  $\mathfrak{DN}(\Sigma_B, \Sigma_S)$ , has five sorts defined as follows:*

$\mathcal{A}(\text{base})$	$t ::=$	$x \mid b \mid (t, t) \mid \text{fst } t \mid \text{snd } t$
$\mathcal{S}(\text{scalar})$	$S ::=$	$x \mid C(\alpha) \mid \delta_{t,t}$
$\mathcal{D}(\text{Dirac notation})$	$D ::=$	$x \mid \mathbf{0} \mid \mathbf{1} \mid D^\dagger \mid D \times D \mid D.D \mid D + D \mid D \cdot D \mid D \otimes D \mid K \mid B \mid O$
$\mathcal{K}(\text{ket})$	$K ::=$	$ s\rangle$
$\mathcal{B}(\text{bra})$	$B ::=$	$\langle s $
$\mathcal{O}(\text{operator})$	$O ::=$	

Here  $x$  is a variable,  $b$  is a term of  $\Sigma_B$  and  $\alpha$  is a term of  $\Sigma_S$ .

[YX] : Here the ordered sorts are only for the purpose of expressing the theory. In **CiME2** and the implementation, there is only one universal sort.

**Remark:** Although we use the same symbol in different sorts ( $B \cdot K$  and  $O \cdot K$ , for example), they are actually different funtions and can be easily distinguished by the context. This is also reflected in the **CiME** script.

[YX] : I encountered some confluence problem for conjugate and transpose, and the natural solution is to consider them as a whole ( $\dagger$ ), which is also more common in quantum computing.

#### 1.2.1 Dirac Notation Algebra

With the language  $\mathfrak{DN}(\Sigma_B, \Sigma_S)$ , we want to define when two terms are equal. This is described by an algebra with axioms.

**BASIS**

$$\begin{aligned}\mathbf{fst} \, (s, t) &= s \\ \mathbf{snd} \, (s, t) &= t \\ (\mathbf{fst} \, s, \mathbf{snd} \, s) &= s\end{aligned}$$

**DOT-DUAL**

$$B \cdot K = K^\top \cdot B^\top$$

**DELTA**

$$\begin{aligned}(\text{DELTA C}) \quad & \delta_{s,t} = \delta_{t,s} \\ (\text{DELTA EQ}) \quad & \frac{(\bigwedge_i s_i =_{\text{BASIS}} t_i) \leftrightarrow (\bigwedge_i s'_i =_{\text{BASIS}} t'_i) \text{ is valid}}{\prod_i \delta_{s_i, t_i} = \prod_i \delta_{s'_i, t'_i}} \\ (\text{DELTA 1}) \quad & \frac{\bigwedge_i s_i =_{\text{BASIS}} t_i \text{ is valid}}{\prod_i \delta_{s_i, t_i} = 1} \\ (\text{DELTA 0}) \quad & \frac{\bigwedge_i s_i =_{\text{BASIS}} t_i \text{ is not satisfiable}}{\prod_i \delta_{s_i, t_i} = 0}\end{aligned}$$

**SCR-COMPLEX**

$$\begin{aligned}a + b &= b + a \\ (a + b) + c &= a + (b + c) \\ a \times b &= b \times a \\ (a \times b) \times c &= a \times (b \times c) \\ C(\alpha) + C(\beta) &= C(\alpha + \beta) \\ C(\alpha) \times C(\beta) &= C(\alpha \times \beta) \\ C(\alpha)^* &= C(\alpha^*) \\ C(0) + a &= a \\ C(0) \times a &= C(0) \\ C(1) \times a &= a \\ a \times (b + c) &= a \times b + a \times c \\ \delta_{s,t}^* &= \delta_{s,t} \\ (a + b)^* &= a^* + b^* \\ (a \times b)^* &= a^* \times b^* \\ (a^*)^* &= a \\ (B \cdot K)^* &= K^\dagger \cdot B^\dagger\end{aligned}$$

**SCR-DOT**

$A \cdot B$  is interpreted as inner product.

$$\begin{aligned}
\mathbf{0} \cdot K &= C(0) \\
B \cdot (S.K) &= S \times (B \cdot K) \\
\langle s | \cdot | t \rangle &= \delta_{s,t} \\
(B_1 \otimes B_2) \cdot (K_1 \otimes K_2) &= (B_1 \cdot K_1) \times (B_2 \cdot K_2)
\end{aligned}$$

**UNI-DOT**

$$\begin{aligned}
\mathbf{0} \cdot D &= \mathbf{0} \\
\mathbf{1}_O \cdot D &= D \\
D_1 \cdot (S.D_2) &= S.(D_1 \cdot D_2) \\
D_0 \cdot (D_1 + D_2) &= D_0 \cdot D_1 + D_0 \cdot D_2 \\
D_0 \cdot (D_1 \cdot D_2) &= (D_0 \cdot D_1) \cdot D_2 \\
(D_1 \otimes D_2) \cdot (D'_1 \otimes D'_2) &= (D_1 \cdot D'_1) \otimes (D_2 \cdot D'_2)
\end{aligned}$$

For example one interpretation for the last axiom is:

$$(O_1 \otimes_O O_2) \cdot_K (K_1 \otimes_K K_2) = (O_1 \cdot_K K_1) \otimes_K (O_2 \cdot_K K_2).$$

**UNI-TSR**

$$\begin{aligned}
\mathbf{0} \otimes D &= \mathbf{0} \\
D \otimes \mathbf{0} &= \mathbf{0} \\
\mathbf{1}_O \otimes \mathbf{1}_O &= \mathbf{1}_O \\
|s\rangle \otimes |t\rangle &= |(s, t)\rangle \\
D_1 \otimes (S.D_2) &= S.(D_1 \otimes D_2) \\
D_0 \otimes (D_1 + D_2) &= D_0 \otimes D_1 + D_0 \otimes D_2
\end{aligned}$$

**ADJ**

$$\begin{aligned}
\mathbf{0}^\dagger &= \mathbf{0} \\
\mathbf{1}_O^\dagger &= \mathbf{1}_O \\
|t\rangle^\dagger &= \langle t| \\
(D^\dagger)^\dagger &= D \\
(S.D)^\dagger &= S^*. (D^\dagger) \\
(D_1 + D_2)^\dagger &= D_1^\dagger + D_2^\dagger \\
(D_1 \cdot D_2)^\dagger &= D_2^\dagger \cdot D_1^\dagger \\
(D_1 \otimes D_2)^\dagger &= D_1^\dagger \otimes D_2^\dagger \\
(K \otimes B)^\dagger &= B^\dagger \otimes K^\dagger
\end{aligned}$$

**LINEAR**

$$\begin{aligned}
D_1 + D_2 &= D_2 + D_1 \\
(D_1 + D_2) + D_3 &= D_1 + (D_2 + D_3) \\
C(0).D &= \mathbf{0} \\
C(1).D &= D \\
D + \mathbf{0} &= D \\
S.\mathbf{0} &= \mathbf{0} \\
S_1.(S_2.D) &= (S_1 \times S_2).D \\
S.(D_1 + D_2) &= S.D_1 + S.D_2
\end{aligned}$$

**OUTER**

$$\begin{aligned}
(K_1 \otimes B) \cdot K_2 &= (B \cdot K_2).K_1 \\
B_1 \cdot (K \otimes B_2) &= (B_1 \cdot K).B_2 \\
(K \otimes B) \cdot O &= K \otimes (B \cdot O) \\
O \cdot (K \otimes B) &= (O \cdot K) \otimes B \\
(K_1 \otimes B_1) \otimes (K_2 \otimes B_2) &= (K_1 \otimes K_2) \otimes (B_1 \otimes B_2)
\end{aligned}$$

**TRANS-UNI**

$$\begin{aligned}
\mathbf{0}^\top &= \mathbf{0} \\
\mathbf{1}_O^\top &= \mathbf{1}_O \\
|s\rangle^\top &= \langle s| \\
(D^\dagger)^\top &= (D^\top)^\dagger \\
(D^\top)^\top &= D \\
(S.D)^\top &= S.D^\top \\
(D_1 + D_2)^\top &= D_1^\top + D_2^\top \\
(D_1 \cdot D_2)^\top &= D_2^\top \cdot D_1^\top \\
(D_1 \otimes D_2)^\top &= D_1^\top \otimes D_2^\top \\
(K \otimes B)^\top &= B^\top \otimes K^\top
\end{aligned}$$

**SUM**

(only part of the equations)

$$\begin{aligned}
\sum_i \sum_j A &= \sum_j \sum_i A \\
\sum_i \sum_j (\langle i|A|j\rangle) \cdot |i\rangle\langle j| &= A \\
&\vdots
\end{aligned}$$

**1.2.2 Semantics**

Constructed in CoqQ.

### 1.2.3 Module Theories

We want to study the theory of Dirac notations modulo the scalars and classical basis. This idea is inspired by the Lineal paper [2].

**Definition 1.2.4** (complex scalar rewrite system). A *complex scalar rewrite system* is a rewrite system  $\mathfrak{S}$  for  $\Sigma_{\mathfrak{S}}$  such that:

- $\mathfrak{S}$  is terminating and ground confluent,
- $+$  and  $\times$  are AC-symbols,
- for all closed terms  $\alpha$ ,  $\beta$  and  $\gamma$ , the pair of terms

- $0 + \alpha$  and  $\alpha$
- $0 \times \alpha$  and  $0$ ,
- $1 \times \alpha$  and  $\alpha$ ,
- $\alpha \times (\beta + \gamma)$  and  $\alpha \times \beta + \alpha \times \gamma$ ,
- $(\alpha + \beta)^*$  and  $\alpha^* + \beta^*$ ,
- $(\alpha \times \beta)^*$  and  $\alpha^* \times \beta^*$ ,
- $(\alpha^*)^*$  and  $\alpha$ ,

have the same normal forms,

- $0$  and  $1$  are normal terms.

**Definition 1.2.5** (atomic basis rewrite system). An *atomic basis rewrite system*  $\mathfrak{B}$  is a terminating and ground confluent rewrite system for  $\Sigma_{\mathfrak{B}}$ .

### 1.2.4 Super Merging

*[YX] : Consider everything as linear operators.*

#### BASIS

$$\vdash \text{fst}(e_1, e_2) \triangleright e_1 \quad \vdash \text{snd}(e_1, e_2) \triangleright e_2 \quad \vdash (\text{fst } e, \text{snd } e) \triangleright e$$

#### DELTA

$$\begin{aligned} \vdash \delta_{u,(s,t)} &\triangleright \delta_{\text{fst } u,s} \times \delta_{\text{snd } u,t} \\ \vdash \delta_{\text{fst } u,\text{fst } v} \times \delta_{\text{snd } u,\text{snd } v} &\triangleright \delta_{u,v} \end{aligned}$$

#### CONJ

$$\begin{aligned} \vdash C(\alpha)^* &\triangleright C(\alpha^*) \\ \vdash \delta_{s,t}^* &\triangleright \delta_{s,t} \\ \vdash (A + B)^* &\triangleright A^* + B^* \\ \vdash (A \times B)^* &\triangleright A^* \times B^* \\ \vdash (A^*)^* &\triangleright A \end{aligned}$$

**ADJ**

$$\begin{aligned}
&\vdash \mathbf{0}^\dagger \triangleright \mathbf{0} \\
&\vdash \mathbf{1}^\dagger \triangleright \mathbf{1} \\
&\vdash \langle t |^\dagger \triangleright |t\rangle \\
&\vdash |t\rangle^\dagger \triangleright \langle t| \\
&\vdash (A^\dagger)^\dagger \triangleright A \\
&\vdash (S.A)^\dagger \triangleright S^*.A^\dagger \\
&\vdash (A+B)^\dagger \triangleright A^\dagger + B^\dagger \\
&\vdash (A \cdot B)^\dagger \triangleright B^\dagger \cdot A^\dagger \\
&\vdash (A \otimes B)^\dagger \triangleright A^\dagger \otimes B^\dagger
\end{aligned}$$

**ADDS**

$$\begin{aligned}
&\vdash C(0) + S \triangleright S \\
&\vdash C(\alpha) + C(\beta) \triangleright C(\alpha + \beta) \\
&\vdash D + D \triangleright C(1 + 1) \times D \\
&\vdash C(a) \times A + A \triangleright C(a + 1) \times A \\
&\vdash C(a) \times A + C(b) \times A \triangleright C(a + b) \times D
\end{aligned}$$

**MLT**

$$\begin{aligned}
&\vdash C(0) \times A \triangleright C(0) \\
&\vdash C(1) \times A \triangleright A \\
&\vdash C(\alpha) \times C(\beta) \triangleright C(\alpha \times \beta) \\
&\vdash A \times (B + C) \triangleright A \times B + A \times C
\end{aligned}$$

**SCR**

$$\begin{aligned}
&\vdash C(0).A \triangleright \mathbf{0} \\
&\vdash A.\mathbf{0} \triangleright \mathbf{0} \\
&\vdash C(1).X \triangleright X \\
&\vdash A.(B.X) \triangleright (A \times B).X \\
&\vdash S.(A + B) \triangleright S.A + S.B
\end{aligned}$$

**ADD**

$$\begin{aligned}
&\vdash \mathbf{0} + A \triangleright A \\
&\vdash D + D \triangleright C(1 + 1).D \\
&\vdash A.D + D \triangleright (A + C(1)).D \\
&\vdash A.D + B.D \triangleright (A + B).D
\end{aligned}$$

**DOT**

$$\begin{aligned}
&\vdash \mathbf{0} \cdot A \triangleright \mathbf{0} \\
&\vdash A \cdot \mathbf{0} \triangleright \mathbf{0} \\
&\vdash \mathbf{1} \cdot A \triangleright A \\
&\vdash A \cdot \mathbf{1} \triangleright A \\
&\vdash \langle s | \cdot | t \rangle \triangleright \delta_{s,t} \\
&\vdash (S.A) \cdot B \triangleright S.(A \cdot B) \\
&\vdash A \cdot (S.B) \triangleright S.(A \cdot B) \\
&\vdash (A + B) \cdot C \triangleright A \cdot C + B \cdot C \\
&\vdash A \cdot (B + C) \triangleright A \cdot B + A \cdot C \\
&\vdash (A \otimes B) \cdot |t\rangle \triangleright (A \cdot |\mathbf{fst} \ t\rangle) \otimes (B \cdot |\mathbf{snd} \ t\rangle) \\
&\vdash \langle t | \cdot (A \otimes B) \triangleright (\langle \mathbf{fst} \ t | \cdot A) \otimes (\langle \mathbf{snd} \ t | \cdot B) \\
&\vdash (A \otimes B) \cdot (C \otimes D) \triangleright (A \cdot C) \otimes (B \cdot D)
\end{aligned}$$

**TSR**

$$\begin{aligned}
&\vdash \mathbf{0} \otimes X \triangleright \mathbf{0} \\
&\vdash X \otimes \mathbf{0} \triangleright \mathbf{0} \\
&\vdash |s\rangle \otimes |t\rangle \triangleright |(s, t)\rangle \\
&\vdash \langle s | \otimes \langle t | \triangleright \langle (s, t) | \\
&\vdash \mathbf{1} \otimes \mathbf{1} \triangleright \mathbf{1} \\
&\vdash (S.A) \otimes B \triangleright S.(A \otimes B) \\
&\vdash A \otimes (S.B) \triangleright S.(A \otimes B) \\
&\vdash (A + B) \otimes C \triangleright A \otimes C + B \otimes C \\
&\vdash A \otimes (B + C) \triangleright A \otimes B + A \otimes C
\end{aligned}$$

**1.2.5 TRS for DN**

**Definition 1.2.6** (TRS  $\mathfrak{DN}$ ). *The AC-rewrite system  $\mathfrak{DN}$  consists of all the rules in Sec.1.2.5. The AC-symbols are  $+$  (for all sorts) and  $\times$ . The commutative symbol is  $\delta_{s,t}$ .*

**BASIS**

$$\vdash \mathbf{fst} \ (e_1, e_2) \triangleright e_1 \quad \vdash \mathbf{snd} \ (e_1, e_2) \triangleright e_2 \quad \vdash (\mathbf{fst} \ e, \mathbf{snd} \ e) \triangleright e$$

**DELTA**

$$\begin{aligned}
&\vdash \delta_{u, (s, t)} \triangleright \delta_{\mathbf{fst} \ u, s} \times \delta_{\mathbf{snd} \ u, t} \\
&\vdash \delta_{\mathbf{fst} \ u, \mathbf{fst} \ v} \times \delta_{\mathbf{snd} \ u, \mathbf{snd} \ v} \triangleright \delta_{u, v}
\end{aligned}$$

**Remark:** These rules are for completion.

**SCR-COP**

$$\begin{aligned}
& \vdash C(0) + S \triangleright S \\
& \vdash C(\alpha) + C(\beta) \triangleright C(\alpha + \beta) \\
& \vdash S + S \triangleright C(1 + 1) \times S \\
& \vdash C(\alpha) \times S + S \triangleright C(\alpha + 1) \times S \\
& \vdash C(\alpha) \times S + C(\beta) \times S \triangleright C(\alpha + \beta) \times S \\
\\
& \vdash C(0) \times S \triangleright C(0) \\
& \vdash C(1) \times S \triangleright S \\
& \vdash C(\alpha) \times C(\beta) \triangleright C(\alpha \times \beta) \\
& \vdash S_1 \times (S_2 + S_3) \triangleright S_1 \times S_2 + S_1 \times S_3 \\
\\
& \vdash C(\alpha)^* \triangleright C(\alpha^*) \\
& \vdash \delta_{s,t}^* \triangleright \delta_{s,t} \\
& \vdash (S_1 + S_2)^* \triangleright S_1^* + S_2^* \\
& \vdash (S_1 \times S_2)^* \triangleright S_1^* \times S_2^* \\
& \vdash (S^*)^* \triangleright S \\
& \vdash (B \cdot K)^* \triangleright K^\dagger \cdot B^\dagger
\end{aligned}$$

**Remark:** We use the symbol  $C$  to denoted the complex scalar in the module theory  $\Sigma_{\mathcal{S}}$ .

**SCR-DOT**

$$\begin{aligned}
& \vdash \mathbf{0} \cdot K \triangleright C(0) \\
& \vdash B \cdot \mathbf{0} \triangleright C(0) \\
& \vdash (S.B) \cdot K \triangleright S \times (B \cdot K) \\
& \vdash B \cdot (S.K) \triangleright S \times (B \cdot K) \\
& \vdash (B_1 + B_2) \cdot K \triangleright B_1 \cdot K + B_2 \cdot K \\
& \vdash B \cdot (K_1 + K_2) \triangleright B \cdot K_1 + B \cdot K_2 \\
& \vdash \langle s | \cdot | t \rangle \triangleright \delta_{s,t} \\
& \vdash (B_1 \otimes B_2) \cdot | t \rangle \triangleright (B_1 \cdot |\mathbf{fst} \ t\rangle) \times (B_2 \cdot |\mathbf{snd} \ t\rangle) \\
& \vdash \langle t | \cdot (K_1 \otimes K_2) \triangleright (\langle \mathbf{fst} \ t | \cdot K_1) \times (\langle \mathbf{snd} \ t | \cdot K_2) \\
& \vdash (B_1 \otimes B_2) \cdot (K_1 \otimes K_2) \triangleright (B_1 \cdot K_1) \times (B_2 \cdot K_2)
\end{aligned}$$

**Remark:** The difficulty here comes from Hilbert space structure. The intuition is that, we decompose the multiplication (inner product) when at least one side is explicitly in tensor product form.

**Remark:** Notice that we don't consider  $\vdash (S.B) \cdot K \triangleright (S^*). (B \cdot K)$ . Inner product is linear (not conjugate linear) on  $B$ , because  $B$  is already in the dual space.

**SCR-SORT**

$$\begin{aligned}
& \vdash (B \cdot O) \cdot K \triangleright B \cdot (O \cdot K) \\
& \vdash \langle s | \cdot ((O_1 \otimes O_2) \cdot K) \triangleright ((\langle \mathbf{fst} \ s | \cdot O_1) \otimes (\langle \mathbf{snd} \ t | \cdot O_2)) \cdot K \\
& \vdash (B_1 \otimes B_2) \cdot ((O_1 \otimes O_2) \cdot K) \triangleright ((B_1 \cdot O_1) \otimes (B_2 \cdot O_2)) \cdot K
\end{aligned}$$

**Remark:** The first rule sorts the multiplication to the right, which breaks the symmetry of ket and bra. The remaining three rules are for completion.



**ADJ-UNI**

$$\begin{aligned}
&\vdash \mathbf{0}^\dagger \triangleright \mathbf{0} \\
&\vdash (X^\dagger)^\dagger \triangleright X \\
&\vdash (S.X)^\dagger \triangleright S^*. (X^\dagger) \\
&\vdash (X_1 + X_2)^\dagger \triangleright X_1^\dagger + X_2^\dagger
\end{aligned}$$

**ADJ-KET**

$$\begin{aligned}
&\vdash \langle t |^\dagger \triangleright |t\rangle \\
&\vdash (B \cdot O)^\dagger \triangleright O^\dagger \cdot B^\dagger \\
&\vdash (B_1 \otimes B_2)^\dagger \triangleright B_1^\dagger \otimes B_2^\dagger
\end{aligned}$$

**ADJ-BRA**

$$\begin{aligned}
&\vdash |t\rangle^\dagger \triangleright \langle t| \\
&\vdash (O \cdot K)^\dagger \triangleright K^\dagger \cdot O^\dagger \\
&\vdash (K_1 \otimes K_2)^\dagger \triangleright K_1^\dagger \otimes K_2^\dagger
\end{aligned}$$

**ADJ-OPT**

$$\begin{aligned}
&\vdash \mathbf{1}_O^\dagger \triangleright \mathbf{1}_O \\
&\vdash (K \otimes B)^\dagger \triangleright B^\dagger \otimes K^\dagger \\
&\vdash (O_1 \cdot O_2)^\dagger \triangleright O_1^\dagger \cdot O_2^\dagger \\
&\vdash (O_1 \otimes O_2)^\dagger \triangleright O_1^\dagger \otimes O_2^\dagger
\end{aligned}$$

**SCR**

$$\begin{aligned}
&\vdash C(0).X \triangleright \mathbf{0} \\
&\vdash C(1).X \triangleright X \\
&\vdash S.\mathbf{0} \triangleright \mathbf{0} \\
&\vdash S_1.(S_2.X) \triangleright (S_1 \times S_2).X \\
&\vdash S.(X_1 + X_2) \triangleright S.X_1 + S.X_2
\end{aligned}$$

**ADD**

$$\begin{aligned}
&\vdash X + \mathbf{0} \triangleright X \\
&\vdash X + X \triangleright C(1+1).X \\
&\vdash S.X + X \triangleright (S + C(1)).X \\
&\vdash S_1.X + S_2.X \triangleright (S_1 + S_2).X
\end{aligned}$$

**TSR**

$$\begin{aligned}
&\vdash \mathbf{0} \otimes X \triangleright \mathbf{0} \\
&\vdash X \otimes \mathbf{0} \triangleright \mathbf{0} \\
&\vdash |s\rangle \otimes |t\rangle \triangleright |(s, t)\rangle \\
&\vdash \langle s| \otimes \langle t| \triangleright \langle (s, t)| \\
&\vdash \mathbf{1}_O \otimes \mathbf{1}_O \triangleright \mathbf{1}_O \\
&\vdash (K_1 \otimes B_1) \otimes (K_2 \otimes B_2) \triangleright (K_1 \otimes K_2) \otimes (B_1 \otimes B_2) \\
&\vdash (S.X_1) \otimes X_2 \triangleright S.(X_1 \otimes X_2) \\
&\vdash X_1 \otimes (S.X_2) \triangleright S.(X_1 \otimes X_2) \\
&\vdash (X_1 + X_2) \otimes X_3 \triangleright X_1 \otimes X_3 + X_2 \otimes X_3 \\
&\vdash X_1 \otimes (X_2 + X_3) \triangleright X_1 \otimes X_2 + X_1 \otimes X_3
\end{aligned}$$

**KET-MLT**

$$\begin{aligned}
&\vdash \mathbf{0} \cdot K \triangleright \mathbf{0} \\
&\vdash O \cdot \mathbf{0} \triangleright \mathbf{0} \\
&\vdash \mathbf{1}_O \cdot K \triangleright K \\
&\vdash (S.O) \cdot K \triangleright S.(O \cdot K) \\
&\vdash O \cdot (S.K) \triangleright S.(O \cdot K) \\
&\vdash (O_1 + O_2) \cdot K \triangleright O_1 \cdot K + O_2 \cdot K \\
&\vdash O \cdot (K_1 + K_2) \triangleright O \cdot K_1 + O \cdot K_2 \\
&\vdash (K_1 \otimes B) \cdot K_2 \triangleright (B \cdot K_2).K_1 \\
&\vdash (O_1 \cdot O_2) \cdot K \triangleright O_1 \cdot (O_2 \cdot K) \\
&\vdash (O_1 \otimes O_2) \cdot ((O'_1 \otimes O'_2) \cdot K) \triangleright ((O_1 \cdot O'_1) \otimes (O_2 \cdot O'_2)) \cdot K \\
&\vdash (O_1 \otimes O_2) \cdot |t\rangle \triangleright (O_1 \cdot |\mathbf{fst} \ t\rangle) \otimes (O_2 \cdot |\mathbf{snd} \ t\rangle) \\
&\vdash (O_1 \otimes O_2) \cdot (K_1 \otimes K_2) \triangleright (O_1 \cdot K_1) \otimes (O_2 \cdot K_2)
\end{aligned}$$

**Remark:** Again, the difficulty comes from space structure. The intuition for reductions is also the same: decompose the multiplication when at least one side is explicitly in tensor product form.

**Remark:** The rules for bra are symmetric to the rules for ket. Only the correspondence of rules in red are different:

$$\begin{aligned}
&\vdash B_1 \cdot (K \otimes B_2) \triangleright (B_1 \cdot K).B_2 \\
&\vdash B \cdot (O_1 \cdot O_2) \triangleright (B \cdot O_1) \cdot O_2 \\
&\vdash (B \cdot (O_1 \otimes O_2)) \cdot (O'_1 \otimes O'_2) \triangleright B \cdot ((O_1 \cdot O'_1) \otimes (O_2 \cdot O'_2))
\end{aligned}$$

**OPT-OUTER**

$$\begin{aligned}
&\vdash \mathbf{0} \otimes B \triangleright \mathbf{0} \\
&\vdash K \otimes \mathbf{0} \triangleright \mathbf{0} \\
&\vdash (S.K) \otimes B \triangleright S.(K \otimes B) \\
&\vdash K \otimes (S.B) \triangleright S.(K \otimes B) \\
&\vdash (K_1 + K_2) \otimes B \triangleright K_1 \otimes B + K_2 \otimes B \\
&\vdash K \otimes (B_1 + B_2) \triangleright K \otimes B_1 + K \otimes B_2
\end{aligned}$$

**OPT-MLT**

$$\begin{aligned}
& \vdash \mathbf{0} \cdot O \triangleright \mathbf{0} \\
& \vdash O \cdot \mathbf{0} \triangleright \mathbf{0} \\
& \vdash \mathbf{1}_O \cdot O \triangleright O \\
& \vdash O \cdot \mathbf{1}_O \triangleright O \\
& \vdash (K \otimes B) \cdot O \triangleright K \otimes (B \cdot O) \\
& \vdash O \cdot (K \otimes B) \triangleright (O \cdot K) \otimes B \\
& \vdash (S.O_1) \cdot O_2 \triangleright S.(O_1 \cdot O_2) \\
& \vdash O_1 \cdot (S.O_2) \triangleright S.(O_1 \cdot O_2) \\
& \vdash (O_1 + O_2) \cdot O_3 \triangleright O_1 \cdot O_3 + O_2 \cdot O_3 \\
& \vdash O_1 \cdot (O_2 + O_3) \triangleright O_1 \cdot O_2 + O_1 \cdot O_3 \\
& \vdash (O_1 \cdot O_2) \cdot O_3 \triangleright O_1 \cdot (O_2 \cdot O_3) \\
& \vdash (O_1 \otimes O_2) \cdot (O'_1 \otimes O'_2) \triangleright (O_1 \cdot O'_1) \otimes (O_2 \cdot O'_2) \\
& \vdash (O_1 \otimes O_2) \cdot ((O'_1 \otimes O'_2) \cdot O_3) \triangleright ((O_1 \cdot O'_1) \otimes (O_2 \cdot O'_2)) \cdot O_3
\end{aligned}$$

**1.3 Confluence of DN**

We use **CiME2** to check the local confluence of the TRS of typed Dirac notation automatically. Because our TRS contains modulo theories and AC-symbols, we need to apply some the techniques:

- consider a smallest instance of the complex scalar rewrite system and utilize the avatar lemma, and
- consider an extended system due to the AC-symbols.

The whole procedure is very similar to that of Lineal.

**1.3.1 Confluence modulo Theory and the Avatar Lemma**

First we introduce some concepts about modularity and the avatar lemma.

**Definition 1.3.1** (Subsumption). *A terminating and confluent relation  $S$  subsumes a relation  $S_0$  if whenever  $t \rightarrow_{S_0} u$ ,  $t$  and  $u$  have the same  $S$ -normal form.*

**Definition 1.3.2** (Commuting relations). *Two relations  $X$  and  $Y$  are said to be commuting if whenever  $t \rightarrow_X u$  and  $t \rightarrow_Y v$ , there exists a term  $w$  such that  $u \rightarrow_Y w$  and  $v \rightarrow_X w$ .*

**Proposition 1.3.1** (The avatar lemma). *[1] Let  $X$ ,  $S$  and  $S_0$  be three relations defined on a set such that:*

- $S$  is terminating and confluent;
- $S$  subsumes  $S_0$ ;
- $S_0 \cup X$  is locally confluent;
- $X$  commutes with  $S^*$ .

*Then, the relation  $S \cup X$  is locally confluent.*

We now present a smallest instantiation  $\mathfrak{S}_0$  of a complex scalar rewrite system.

**Definition 1.3.3** (The rewrite system  $\mathfrak{S}_0$ ). *The system  $\mathfrak{S}_0$  is defined by the following rules:*

$$\begin{aligned}
0 + \alpha &\triangleright \alpha \\
0 \times \alpha &\triangleright 0 \\
1 \times \alpha &\triangleright \alpha \\
\alpha \times (\beta + \gamma) &\triangleright \alpha \times \beta + \alpha \times \gamma \\
0^* &\triangleright 0 \\
1^* &\triangleright 1 \\
(\alpha + \beta)^* &\triangleright \alpha^* + \beta^* \\
(\alpha \times \beta)^* &\triangleright \alpha^* \times \beta^* \\
(\alpha^*)^* &\triangleright \alpha
\end{aligned}$$

where  $+$  and  $\times$  are AC-symbols.

### 1.3.2 AC-rewriting and Extension Rules

If a TRS  $R$  contains AC-symbols, checking the critical pairs of  $R$  is not enough to ensure the local confluence of  $R$ . This is because the LHS of rules can match the term unlocally. For example:

$$(K_0 + K_1) + K_0 \triangleright C(1 + 1).K_0 + K_1$$

by the rule  $\vdash K + K \triangleright C(1 + 1).K$ . We need to consider an extended system, as indicated by the following definition.

**Definition 1.3.4** (The extension rules). *Let  $X$  be a AC-rewrite system with AC symbols  $f_1, \dots, f_n$ . We define the AC-rewrite system  $X_{ext}$  as containing the same AC symbols as  $X$ , the same rules as  $X$ , plus the rule  $f_i(t, x) \rightarrow f_i(u, x)$  for each rule  $t \rightarrow u$  of  $X$  where the head symbol of  $t$  is  $f_i$ .*

**Proposition 1.3.2.** *The globally AC reduction relation of system  $X$  is confluent if and only if the locally AC reduction relation of  $X_{ext}$  is confluent.*

And we can check the local confluence of locally AC reduction of  $X_{ext}$  by calculating the critical pairs.

**Proposition 1.3.3** ( $\mathfrak{S}_{0ext}$ ). *The system  $\mathfrak{S}_{0ext}$  is formed by the rules in  $S_0$  and the rule*

$$\begin{aligned}
(0 + \alpha) + \chi &\triangleright \alpha + \chi \\
(0 \times \alpha) \times \chi &\triangleright 0 \times \chi \\
(1 \times \alpha) \times \chi &\triangleright \alpha \times \chi \\
(\alpha \times (\beta + \gamma)) \times \chi &\triangleright (\alpha \times \beta + \alpha \times \gamma) \times \chi
\end{aligned}$$

**Proposition 1.3.4.** *The system  $\mathfrak{DN}_{ext}$  is formed by the rules in  $\mathfrak{DN}$  and the rule*

$$\begin{aligned}
& (C(0) + a) + x \triangleright a + x \\
& (C(a) + C(b)) + x \triangleright C(a + b) + x \\
& (S + S) + x \triangleright (C(1 + 1) \times S) + x \\
& ((C(a) \times S) + S) + x \triangleright (C(a + 1) \times S) + x \\
& ((C(a) \times S) + (C(b) \times S)) + x \triangleright (C(a + b) \times S) + x \\
\\
& (C(0) \times a) \times x \triangleright C(0) \times x \\
& (C(1) \times a) \times x \triangleright a \times x \\
& (C(a) \times C(b)) \times x \triangleright C(a \times b) \times x \\
& (S_1 \times (S_2 + S_3)) \times x \triangleright ((S_1 \times S_2) + (S_1 \times S_3)) \times x \\
\\
& (X + \mathbf{0}_K) + x \triangleright X + x \\
& (X + X) + x \triangleright C(1 + 1).X + x \\
& (S.X + X) + x \triangleright (S + 1).X + x \\
& (S_1.X + S_2.X) + x \triangleright (S_1 + S_2).X + x
\end{aligned}$$

*[YX] : These part is not encoded in CiME2 code because the tool is already AC-extended.*

**Proposition 1.3.5.** *The system  $\mathfrak{S}_{0ext} \cup \mathfrak{DN}_{ext}$  is locally confluent.*

*Proof.* Checked by CiME2. □

### 1.3.3 Checking by CiME

We encoded the typed Dirac notation in a simple TRS and checked the confluence of the whole system. Here is a summary of the rules.

Type	Rule Number
type checking	27
overloading polymorphic symbols	22
complex scalar avatar	9
Dirac notation	141
AC-symbol extension rules	25
Total	224

And all the 3133 critical pairs are joinable.

#### About Redundancy

I also checked through a Python script that removing any of the reduction rule will lead to non-confluence or change of the equational theory. For example, I found that removing

$$\vdash (\mathbf{fst} \ e, \mathbf{snd} \ e) \triangleright e$$

or

$$\vdash (K_1 \otimes B_1) \otimes (K_2 \otimes B_2) \triangleright (K_1 \otimes K_2) \otimes (B_1 \otimes B_2)$$

respectively will still lead to a confluence TRS, but the equivalence induced by the TRS will be changed.

## 1.4 Termination of DN

Termination tool: see [https://link.springer.com/chapter/10.1007/978-3-030-17502-3\\_10](https://link.springer.com/chapter/10.1007/978-3-030-17502-3_10)

May be we can try MU-TERM.

## 1.5 Extension: Delta\*

In the backbone rewriting system DN, we don't attempt to reduce Delta operators  $\delta_{s,t}$  to constants 0 or 1. This is because the complete reduction needs some semantic analysis and can be quite difficult. This section tries to find such a good extension.

To demonstrate the difficulty, we first consider the following example:

$$\delta_{i,j} \times \delta_{i,k} = \delta_{i,j} \times \delta_{j,k}.$$

This equality holds, but it is hard to detect merely by the syntax. In general, The equality and reduction of Delta operators need the semantic analysis, which is presented below.

### DELTA-COMPLETE

$$\begin{array}{l} \text{(DELTAEQ)} \quad \frac{(\bigwedge_i s_i =_{\text{BASIS}} t_i) \leftrightarrow (\bigwedge_i s'_i =_{\text{BASIS}} t'_i) \text{ is valid}}{\prod_i \delta_{s_i, t_i} = \prod_i \delta_{s'_i, t'_i}} \\ \\ \text{(DELTA1)} \quad \frac{\bigwedge_i s_i =_{\text{BASIS}} t_i \text{ is valid}}{\prod_i \delta_{s_i, t_i} \triangleright 1} \\ \\ \text{(DELTA0)} \quad \frac{\bigwedge_i s_i =_{\text{BASIS}} t_i \text{ is not satisfiable}}{\prod_i \delta_{s_i, t_i} \triangleright 0} \end{array}$$

*[YX] : We need to prove that the backbone system together with these rules are confluent.*

### 1.5.1 Extension for Delta Operator (simple version)

It's impractical to fully decide the three conditions, espacially when the atomic basis  $\mathfrak{B}$  is complicated. Therefore we present here a simpler extension to rewrite these Delta operators in common cases.

### DELTA\*

$$\begin{array}{l} \text{(DELTA1)} \quad \vdash \delta_{s,s} \triangleright 1 \\ \\ \text{(DELTA0)} \quad \frac{s, t \text{ are different closed terms}}{\vdash \delta_{s,t} \triangleright 0} \end{array}$$

### 1.5.2 Extension for Delta Operator (SMT solver)

We can use a SMT solver to check the conditions for the equality **DeltaEQ** and rules in DELTA-COMPLETE.

## 1.6 Extension: Sum and Transpose

**Definition 1.6.1** (Set).

$$S ::= \emptyset \mid \mathbf{U} \mid S \cup S$$

### SET-SIMP

$$\begin{array}{l} \emptyset \cup S \triangleright S \\ \mathbf{U} \cup S \triangleright \mathbf{U} \end{array}$$

**Definition 1.6.2** (DNE). *The **language of Dirac Notation Extended**, denoted as  $\mathfrak{DNE}(\Sigma_B, \Sigma_S)$ , consists of the same sorts as  $\mathfrak{DN}(\Sigma_B, \Sigma_S)$  with the following new symbols:*

$$D ::= D^\top \mid \sum_{i \in M}^{(S)} D \mid \sum_{i \in M} D \mid \lambda x. D \mid D D$$

*The index  $i$  for summation is a bind variable, and  $M$  is the variable for the multiset of summation. The common considerations for bind variables are applied here:  $\alpha$ -conversion and substitutions.*

*[YX] : The biggest problem with variable set is that we cannot specify whether the term belongs to the set or not. Consider the example:  $\sum_{i \in M} \delta_{i,x}$ , this term reduces to 1 when  $x \in M$  and 0 otherwise.*

## 1.7 TRS for DNE

**Definition 1.7.1** (TRS  $\mathfrak{DNE}$ ). *The AC-rewrite system  $\mathfrak{DNE}$  consists of all the rules in  $\mathfrak{DN}$  as well as the rules in this section.*

### TRANS-UNI

$$\begin{aligned} \mathbf{0}^\top &\triangleright \mathbf{0} \\ (X^\dagger)^\top &\triangleright (X^\top)^\dagger \\ (X^\top)^\top &\triangleright X \\ (S.X)^\top &\triangleright S.X^\top \\ (X_1 + X_2)^\top &\triangleright X_1^\top + X_2^\top \end{aligned}$$

### TRANS-KET

$$\begin{aligned} \langle s |^\top &\triangleright |s\rangle \\ (B \cdot O)^\top &\triangleright O^\top \cdot B^\top \\ (B_1 \otimes B_2)^\top &\triangleright B_1^\top \otimes B_2^\top \end{aligned}$$

### TRANS-BRA

$$\begin{aligned} |s\rangle^\top &\triangleright \langle s| \\ (O \cdot K)^\top &\triangleright K^\top \cdot O^\top \\ (K_1 \otimes K_2)^\top &\triangleright K_1^\top \otimes K_2^\top \end{aligned}$$

### TRANS-OPT

$$\begin{aligned} \mathbf{1}_O^\top &\triangleright \mathbf{1}_O \\ (K \otimes B)^\top &\triangleright B^\top \otimes K^\top \\ (O_1 \cdot O_2)^\top &\triangleright O_2^\top \cdot O_1^\top \\ (O_1 \otimes O_2)^\top &\triangleright O_1^\top \otimes O_2^\top \end{aligned}$$

**Remark:** It has been checked by CiME2 that for the TRS  $\mathfrak{DNE}$  extended with transpose rules, the only unjoinable critical pairs are joinable with the equational theory  $B \cdot K = K^\top \cdot B^\top$ .

**SUM-ELIM**

$$\sum_i C(0) \triangleright C(0)$$

$$\sum_i \mathbf{0} \triangleright \mathbf{0}$$

One common condition is attached to all the following rules: variable  $i$  does not have free appearance in term  $s$ . Here  $S.A$  is interpreted as three cases:  $S.K$ ,  $S.B$  and  $S.O$ .

$$\sum_i \delta_{i,s} \triangleright C(1)$$

$$\sum_i (\delta_{i,s} \times S) \triangleright S[i := s]$$

$$\sum_i (\delta_{i,s}.A) \triangleright A[i := s]$$

$$\sum_i ((\delta_{i,s} \times S).A) \triangleright S[i := s].A[i := s]$$

*[YX] : Terms like  $\sum_i \delta_{i,j}$  are not simplified.*

*[YX] : In principle we will want to count the “times” the two indices be equivalent to each other. For example,*

$$\sum_i \delta_{i,i^2} \triangleright 2$$

*But in general this is not possible. But we can limit the case to “ $i$  is the bind variable and  $i$  is not a free variable in expression  $j$ ”.*

The following rules are for ket, bra and operators respectively.

$$\sum_i (\langle i| \cdot K).|i\rangle \triangleright K$$

$$\sum_i (B \cdot |i\rangle).\langle i| \triangleright B$$

$$\sum_i \sum_j (\langle i|A|j\rangle).|i\rangle\langle j| \triangleright A$$

$$\sum_i \sum_j (\langle i|A|j\rangle).|j\rangle\langle i| \triangleright A^\top$$

*[YX] : One may expect these rules to be applied in broader cases, for example:*

$$\sum_i \sum_j (\langle i|A|j\rangle).|j\rangle\langle i| \triangleright A \cdot B,$$

*So we have to pull  $B$  out.*

**SUM-DIST**

Here the zero operator  $\mathbf{0}$  is interpreted in three sorts: ket, bra and operator. The same for  $A^\dagger$ ,  $A^\top$ ,  $S.A$  and addition.

Here  $A \cdot B$  is interpreted as four cases:  $B \cdot K$ ,  $O \cdot K$ ,  $B \cdot O$  and  $O_1 \cdot O_2$ .  $A \otimes B$  is interpreted as four cases:  $K_1 \otimes K_2$ ,  $B_1 \otimes B_2$ ,  $K \otimes B$  and  $O_1 \otimes O_2$ .



$$\begin{aligned}
\sum_i (S_1 \times X) &\triangleright (\sum_i S_1) \times X \\
(\sum_i S)^* &\triangleright \sum_i S^* \\
(\sum_i A)^\dagger &\triangleright \sum_i A^\dagger \\
(\sum_i A)^\top &\triangleright \sum_i A^\top \\
\sum_i (X.A) &\triangleright X.(\sum_i A) \\
\sum_i (S.X) &\triangleright (\sum_i S).X \\
\sum_i (A \cdot X) &\triangleright (\sum_i A) \cdot X \\
\sum_i (X \cdot B) &\triangleright X \cdot (\sum_i B) \\
\sum_i (A \otimes X) &\triangleright (\sum_i A) \otimes X \\
\sum_i (X \otimes B) &\triangleright X \otimes (\sum_i B)
\end{aligned}$$

**Remark:** In the end we choose to pull everything out. The considerations behinds are:

- make SUM-ELIM-7 to SUM-ELIM-10 more complete.

### About **SUM-DIST**: push in or pull out?

We can flip the direction of the rules in **SUM-DIST** to pull elements out of the summation. I believe the difference lies in the **information of bind variable dependences**.

When pushing inside, the information of bind variable independence is lost. This makes elimination rule less complete. Maybe we want to do in the opposite way: pull everything out as much as possible. Those rules involve checking whether the subterm to be pulled out is independent on the bind variable. But this strategy makes matching elimination rule simpler.

### **SUM-COMP**

These rules are for completion.

$$\frac{A \cdot B \triangleright C \quad (\text{bind variable renamed})}{A \cdot (\sum_i B) \triangleright \sum_i C}$$

$$\frac{A \cdot B \triangleright C \quad (\text{bind variable renamed})}{(\sum_i A) \cdot B \triangleright \sum_i C}$$

$$\frac{A \otimes B \triangleright C \quad (\text{bind variable renamed})}{A \otimes (\sum_i B) \triangleright \sum_i C}$$

$$\frac{A \otimes B \triangleright C \quad (\text{bind variable renamed})}{(\sum_i A) \otimes B \triangleright \sum_i C}$$

$$\frac{S \neq C(1)}{\sum_i^{(S)} S \triangleright (\sum_i^{(S)} C(1)) \times S}$$

$$\sum_i A \triangleright (\sum_i^{(S)} C(1)).A$$

*[YX] : But I found a speical critical pair:*

$$\left( \sum_i (\langle i | \cdot K0) (X \otimes |i\rangle) \right) \triangleleft X \otimes \sum_i (\langle i | \cdot K0).|i\rangle \triangleright X \otimes K0$$

*So I guess the completion rules should be further modified.*

## SUM-ADD

Here the addition  $A + B$  is interpreted in four sorts: scalar, ket, bra and operator.

$$(\sum_i A) + (\sum_j B) \triangleright \sum_k (A[i := k] + B[j := k])$$

*[YX] : But maybe we want to rewrite in the opposite direction?*

## 1.8 Problems & Techniques for $\mathfrak{DNE}$

### 1.8.1 Big-op: merge or expand?

#### Problem

We have the equational theory:

$$\sum_{i \in s_0, s_1, \dots, s_n} A = A[i := s_0] + A[i := s_1] + \dots + A[i := s_n]$$

But which side should be the canonical form?

The conservative choice is the right hand side, and the problem is it can be quite inefficient. The problem with left hand side is that sometimes it will lead to unjoinable normal terms. For examle:

$$\sum_{i \in 2} |0\rangle\langle i| + |1\rangle\langle 0| \triangleleft |0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| \triangleright \sum_{i \in 2} |i\rangle\langle 0| + |0\rangle\langle 1|$$

**Solution**

I think the proper solution is to leave the choice to the user. We can use LHS by default and implement a transformation to expand the summation.

**1.8.2 Equivalence checking in Wolfram Engine****Problem**

The module for complex scalars and atomic basis are assumed to be completed. That is, semantical equivalence is identical to having the same syntax. But this is not always true in Wolfram language: terms having different normal forms can be mathematically equivalent. Here is an example:

$e^{i(a+b)}$  and  $e^{ia+ib}$  are equivalent in maths but they have different normal forms in Wolfram language. Nevertheless, this can still be checked by computing `FullSimplify[ $e^{i(a+b)} == e^{ia+ib}$ ]` and this will be reduced to `True`.

**Solution**

We adopt a unique table for Wolfram expressions and search in the table whenever we create a new Wolfram expression. This ensures that semantically equivalent terms only have one instance in the system. While this implementation is inefficient during rewriting, we only transform the normal forms using unique scalar tables when comparing equivalence.

**1.8.3 DOT-DUAL equational theory****Problem**

The TRS  $\mathfrak{DNE}$  is not complete for DOT-DUAL equational theory.

The DOT-DUAL equational theory is expressed as:

$$B \cdot K = K^\top \cdot B^\top.$$

We encountered difficulty in finding the canonical form. Our solution is to define an internal commutative symbol JUX to juxtapose the two possible expressions, with the following rule:

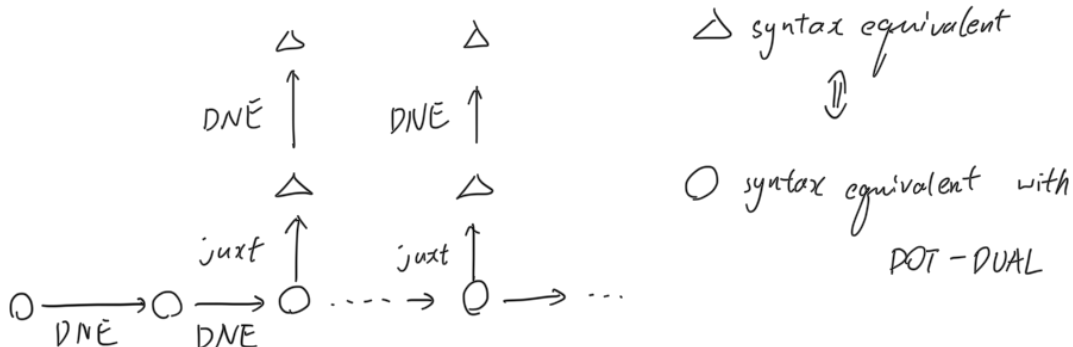
$$(\text{DOT-DUAL-JUX}) \quad B \cdot K \triangleright \text{JUX}(B \cdot K, K^\top \cdot B^\top).$$

$B \cdot K$  and  $K^\top \cdot B^\top$  will be joinable with this rule. But there are two problems: it is not terminating, and its hard to prove confluence. The second problem comes from the fact that normally the two subterms  $A, B$  in  $\text{JUX}(A, B)$  are semantically equivalent, but the standard framework of term rewriting does not support such constraint.

**Solution**

We introduce a new framework called “term rewriting with multiple sections” to separate the terms for rewriting calculation and equivalence comparison. There are two main points:

1. Rewrite terms using  $\mathfrak{DNE}$ .
2. When comparing equivalence extended with DOT-DUAL, we transform the normal term of  $\mathfrak{DNE}$  using `juxt`, which apply the rule DOT-DUAL-JUX iteratively once, and then rewrite the term again using  $\mathfrak{DNE}$ .



### 1.8.4 SUM-SWAP equational theory

#### Problem

The TRS  $\mathfrak{DNE}$  is not complete for SUM-SWAP equational theory, which is:

$$\sum_i \sum_j A = \sum_j \sum_i A.$$

This equational theory can be understood as the result from AC property of addition. Again, we cannot find the canonical form to join the terms on the two sides.

#### Solution

Similar to the techniques of DOT-DUAL-JUX, we introduce another internal AC symbol SUMEQ, with the following rule:

$$(\text{SUM-SWAP-JUX}) \quad \sum_i \sum_j A \triangleright (\sum_i \sum_j A) \text{SUMEQ} (\sum_j \sum_i A)$$

Then we introduce another rewriting section, which first apply the rule SUM-SWAP-JUX once, and then rewrite again using  $\mathfrak{DNE}$ . The syntactic equivalence after this transformation should be equivalent to the equivalence of  $\mathfrak{DNE}$  normal terms with SUM-SWAP equational theory.

### 1.8.5 SUM-ENTRY-EQ equational theory

#### Problem

Consider the equational theory:

$$\begin{aligned} \sum_i ((|i| \cdot K) \cdot |i\rangle) &= K \\ \sum_i (B \cdot |i\rangle) \cdot \langle i| &= B \\ \sum_i \sum_j ((\langle i|A|j\rangle) \cdot |i\rangle \langle j|) &= A \end{aligned}$$

We can rewrite towards the right, but in some scenarios we want to explicitly expand the ket/bra/operators in calculation.

#### Solution

We can let the user to manually control whether they want to rewrite towards the left or the right hand side.

**Conjecture 1.** *Iteratively doing the expansion once is enough for proving the equivalence.*

### 1.8.6 Unjoinable Critical Pair

#### Problem

Unjoinable pair:

$$\sum_i ((\delta_{\mathbf{fst}(i),s} \times \delta_{\mathbf{snd}(i),t}) \cdot |i\rangle) \triangleleft \sum_i (\delta_{i,(s,t)} \cdot |i\rangle) \triangleright |(s,t)\rangle$$

#### Solution

Always put SUM rules in front of others?

Although it does not fix the confluence problem, it's still useful in practice, because soundness still holds. As long as terms are equivalent under some rewriting strategy then they are equivalent in semantics.

## 1.9 Typed Dirac Notation

Now we build a type system based on the untyped Dirac notation. With the typing information, we are able to achieve the polymorphism of the universal application symbol, which is close to how Dirac notation is dealt with in practice.

### Polymorphism of Universal Application

I believe the essence of Dirac notation is the polymorphism of universal application, which means that the same symbol is overloaded as different funtions based on the type of operands. We often use a universal application binary symbol (denoted as  $\circ$ ) to concatenate subterms. For example,

$$\langle a | \circ | b \rangle \circ \langle c | \circ | d \rangle = \langle a | b \rangle \langle c | d \rangle.$$

This term has no ambiguity because universal application is expected to be associative. And  $\circ$  is overloaded differently based on how we associate the calculation. For example,

$$(\langle a | \circ | b \rangle) \circ (\langle c | \circ | d \rangle) = \langle a | b \rangle \times \langle c | d \rangle$$

$$\langle a | \circ (| b \rangle \circ \langle c |) \circ | d \rangle = \langle a | \cdot | b \rangle \langle c | \cdot | d \rangle$$

The complete list of overloading is given by the table:

$X \circ Y$	$Y \in \mathcal{S}$	$Y \in \mathcal{K}$	$Y \in \mathcal{B}$	$Y \in \mathcal{O}$
$X \in \mathcal{S}$	$S_X \times S_Y$	$S_X \cdot K_Y$	$S_X \cdot B_Y$	$S_X \cdot O_Y$
$X \in \mathcal{K}$	$S_Y \cdot K_X$	$K_X \otimes K_Y$	$K_X \otimes B_Y$	
$X \in \mathcal{B}$	$S_Y \cdot B_X$	$B_X \cdot K_Y$	$B_X \otimes B_Y$	$B_X \cdot O_Y$
$X \in \mathcal{O}$	$S_Y \cdot O_X$	$O_X \cdot K_Y$		$O_X \cdot O_Y$

A major topic of this work is to **prove the associativity of the universal application symbol** constructed in the TRS.

### Polymorphism of otimes and cdot

Similar to the universal application symbol  $\circ$ , we also use  $A \otimes B$  and  $A \cdot B$  in a polymorphic way. This is also encoded in the CiME script.

### Type Definition

**Definition 1.9.1** (Types). *The type for basis is defined by*

$$\tau ::= T \mid (\tau * \tau),$$

Here  $T$  is the type for atomic bases. The type for scalar, ket, bra and operator are defined by  $\mathcal{S}$ ,  $\mathcal{K}(\tau)$ ,  $\mathcal{B}(\tau)$  and  $\mathcal{O}(\tau, \sigma)$ , respectively.

#### 1.9.1 typing rules

A typing assumption has the form  $x : \tau$ , meaning variable  $x$  has the type  $\tau$ . A typing context  $\Gamma$  consists of typing assumptions and each variable appears only once.

A typing judgement  $\vdash e : \sigma$  indicates that  $e$  is a term of type  $\sigma$  in context  $\Gamma$ . The well-typed terms are defined by the following rules.

### Typing of Basis

$$\frac{x : \sigma \in \Gamma}{\vdash x : \sigma} \quad \vdash b : T$$

$$\frac{\vdash t_1 : \tau \quad \vdash t_2 : \sigma}{\vdash (t_1, t_2) : (\tau * \sigma)}$$

$$\frac{\vdash t : (\tau * \sigma)}{\vdash \mathbf{fst} \, t : \tau} \quad \frac{\vdash t : (\tau * \sigma)}{\vdash \mathbf{snd} \, t : \sigma}$$

### Typing of Scalar

$$\frac{x : \mathcal{S} \in \Gamma}{\vdash x : \mathcal{S}} \quad \vdash C(\alpha) : \mathcal{S} \quad \frac{\vdash s : \tau \quad \vdash t : \tau}{\vdash \delta_{s,t} : \mathcal{S}}$$

$$\frac{\vdash S_1 : \mathcal{S} \quad \vdash S_2 : \mathcal{S}}{\vdash S_1 + S_2 : \mathcal{S}} \quad \frac{\vdash S_1 : \mathcal{S} \quad \vdash S_2 : \mathcal{S}}{\vdash S_1 \times S_2 : \mathcal{S}}$$

$$\frac{\vdash S : \mathcal{S}}{\vdash S^* : \mathcal{S}}$$

$$\frac{\vdash B : \mathcal{B}(\tau) \quad \vdash K : \mathcal{K}(\tau)}{\vdash B \cdot K : \mathcal{S}}$$

### Typing of Ket

$$\frac{x : \mathcal{K}(\tau) \in \Gamma}{\vdash x : \mathcal{K}(\tau)} \quad \vdash \mathbf{0}_{\mathcal{K}(\tau)} : \mathcal{K}(\tau) \quad \frac{\vdash t : \tau}{\vdash |t\rangle : \mathcal{K}(\tau)}$$

$$\frac{\vdash B : \mathcal{B}(\tau)}{\vdash B^\dagger : \mathcal{K}(\tau)} \quad \frac{\vdash S : \mathcal{S} \quad \vdash K : \mathcal{K}(\tau)}{\vdash S.K : \mathcal{K}(\tau)}$$

$$\frac{\vdash K_1 : \mathcal{K}(\tau) \quad \vdash K_2 : \mathcal{K}(\tau)}{\vdash K_1 + K_2 : \mathcal{K}(\tau)}$$

$$\frac{\vdash O : \mathcal{O}(\tau, \sigma) \quad \vdash K : \mathcal{K}(\sigma)}{\vdash O \cdot K : \mathcal{K}(\tau)}$$

$$\frac{\vdash K_1 : \mathcal{K}(\tau_1) \quad \vdash K_2 : \mathcal{K}(\tau_2)}{\vdash K_1 \otimes K_2 : \mathcal{K}(\tau_1 * \tau_2)}$$

**Typing of Bra**

$$\begin{array}{c}
\frac{x : \mathcal{B}(\tau) \in \Gamma}{\vdash x : \mathcal{B}(\tau)} \quad \vdash \mathbf{0}_{\mathcal{B}(\tau)} : \mathcal{B}(\tau) \quad \frac{\vdash t : \tau}{\vdash \langle t \rangle : \mathcal{B}(\tau)} \\
\\
\frac{\vdash K : \mathcal{K}(\tau)}{\vdash K^\dagger : \mathcal{B}(\tau)} \quad \frac{\vdash S : \mathcal{S} \quad \vdash B : \mathcal{B}(\tau)}{\vdash S.B : \mathcal{B}(\tau)} \\
\\
\frac{\vdash B_1 : \mathcal{B}(\tau) \quad \vdash B_2 : \mathcal{B}(\tau)}{\vdash B_1 + B_2 : \mathcal{B}(\tau)} \\
\\
\frac{\vdash B : \mathcal{B}(\tau) \quad \vdash O : \mathcal{O}(\tau, \sigma)}{\vdash B \cdot O : \mathcal{B}(\sigma)} \\
\\
\frac{\vdash B_1 : \mathcal{B}(\tau_1) \quad \vdash B_2 : \mathcal{B}(\tau_2)}{\vdash B_1 \otimes B_2 : \mathcal{B}(\tau_1 * \tau_2)}
\end{array}$$

**Typing of Operator**

$$\begin{array}{c}
\frac{x : \mathcal{O}(\tau, \sigma) \in \Gamma}{\vdash x : \mathcal{O}(\tau, \sigma)} \quad \vdash \mathbf{0}_{\mathcal{O}(\tau, \sigma)} : \mathcal{O}(\tau, \sigma) \quad \vdash \mathbf{1}_{\mathcal{O}(\tau)} : \mathcal{O}(\tau, \tau) \quad \frac{\vdash K : \mathcal{K}(\tau) \quad \vdash B : \mathcal{B}(\sigma)}{\vdash K \otimes B : \mathcal{O}(\tau, \sigma)} \\
\\
\frac{\vdash O : \mathcal{O}(\tau, \sigma)}{\vdash O^\dagger : \mathcal{O}(\sigma, \tau)} \quad \frac{\vdash S : \mathcal{S} \quad \vdash O : \mathcal{O}(\tau, \sigma)}{\vdash S.O : \mathcal{O}(\tau, \sigma)} \\
\\
\frac{\vdash O_1 : \mathcal{O}(\tau, \sigma) \quad \vdash O_2 : \mathcal{O}(\tau, \sigma)}{\vdash O_1 + O_2 : \mathcal{O}(\tau, \sigma)} \\
\\
\frac{\vdash O_1 : \mathcal{O}(\tau, \sigma) \quad \vdash O_2 : \mathcal{O}(\sigma, \rho)}{\vdash O_1 \cdot O_2 : \mathcal{O}(\tau, \rho)} \\
\\
\frac{\vdash O_1 : \mathcal{O}(\tau_1, \sigma_1) \quad \vdash O_2 : \mathcal{O}(\tau_2, \sigma_2)}{\vdash O_1 \cdot O_2 : \mathcal{O}(\tau_1 * \tau_2, \rho_1 * \rho_2)}
\end{array}$$

**Remark:** The type system is transformed and encoded in the simplest TRS. All the typing rules are treated as reduction rules.

**1.10 Labelled Dirac Notation**

**Definition 1.10.1** (quantum registers).

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

*[YX] : We still have a very speical algorithm deciding a “belonging to” relation.*

**Definition 1.10.2** (register set).

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

Here  $\cup$  is an AC symbol.

**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**

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

**RSET**

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

$$\frac{R_1 \text{ in } R_2}{\{R_1\} \cup \{R_2\} \triangleright \{R_2\}}$$

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

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

$$\frac{R_1 \parallel R_2}{\{R_1\} \setminus \{R_2\} \triangleright \{R_1\}}$$

**Definition 1.10.3** (labelled core language).

$$D ::= D_R \mid D_{R;R} \mid D \tilde{\otimes} D$$

In other words, we don't allow variables for labelled core language. Here,  $\otimes_K$ ,  $\otimes_B$  and  $\otimes_P$  are AC symbols.

**Remark:** We use  $D_R$  in an ambiguous way: the ket/bra or the operator.

**TSR-DECOMP**

$$\begin{aligned} |(s, t)\rangle_{(Q,R)} &\triangleright |s\rangle_Q \otimes |t\rangle_R \\ \langle (s, t)|_{(Q,R)} &\triangleright \langle s|_Q \otimes \langle s|_R \\ \mathbf{0}_{(Q,R)} &\triangleright \mathbf{0}_Q \otimes \mathbf{0}_R \\ \mathbf{0}_{(Q_1, Q_2); R} &\triangleright \mathbf{0}_{Q_1; \mathbf{fst} R} \otimes \mathbf{0}_{Q_2; \mathbf{fst} R} \\ \mathbf{0}_{Q; (R_1, R_2)} &\triangleright \mathbf{0}_{\mathbf{fst} Q; R_1} \otimes \mathbf{0}_{\mathbf{snd} Q; R_2} \\ \mathbf{1}_{(Q,R)} &\triangleright \mathbf{1}_Q \otimes \mathbf{1}_R \\ \mathbf{1}_{(Q_1, Q_2); R} &\triangleright \mathbf{1}_{Q_1; \mathbf{fst} R} \otimes \mathbf{1}_{Q_2; \mathbf{fst} R} \\ \mathbf{1}_{Q; (R_1, R_2)} &\triangleright \mathbf{1}_{\mathbf{fst} Q; R_1} \otimes \mathbf{1}_{\mathbf{snd} Q; R_2} \\ (A \otimes B)_{(Q,R)} &\triangleright A_Q \otimes B_R \\ (A \otimes_O B)_{(Q_1, Q_2); R} &\triangleright A_{Q_1; \mathbf{fst} R} \otimes_O B_{Q_2; \mathbf{snd} R} \\ (A \otimes_O B)_{Q; (R_1, R_2)} &\triangleright A_{\mathbf{fst} Q; R_1} \otimes_O B_{\mathbf{snd} Q; R_2} \end{aligned}$$



**TSR-COMP**

$$\begin{aligned} A_{fst\ Q;fst\ R} \otimes B_{snd\ Q;snd\ R} &\triangleright (A \otimes B)_{Q;R} \\ A_{fst\ R} \otimes B_{snd\ R} &\triangleright (A \otimes B)_R \end{aligned}$$

**DOT-TSR**

$$\frac{R \parallel S}{A_{Q;R} \cdot_O B_{S;T} \triangleright A_{Q;R} \otimes_O B_{S;T}}$$

**LABEL-LIFT**

$$\begin{aligned} (A_R)^\dagger &\triangleright (A^\dagger)_R \\ (A_{Q;R})^\dagger &\triangleright (A^\dagger)_{Q;R} \\ (A_R)^\top &\triangleright (A^\top)_R \\ (A_{Q;R})^\top &\triangleright (A^\top)_{Q;R} \\ (S.A)_R &\triangleright S.(A_R) \\ (S.A)_{Q;R} &\triangleright S.(A_{Q;R}) \\ (A+B)_R &\triangleright A_R + B_R \\ (A+B)_{Q;R} &\triangleright A_{Q;R} + B_{Q;R} \\ A_{Q;R} \cdot_O B_{R;S} &\triangleright (A \cdot_O B)_{Q;S} \\ A_{Q;R} \cdot_O B_R &\triangleright (A \cdot_O B)_{Q;R} \\ A_Q \cdot_O B_{Q;R} &\triangleright (A \cdot_O B)_{Q;R} \\ A_{Q;R} \cdot_K B_R &\triangleright (A \cdot_K B)_Q \\ A_Q \cdot_B B_{Q;R} &\triangleright (A \cdot_B B)_R \\ A_R \cdot B_R &\triangleright (A \cdot B)_R \\ A_R \cdot_I B_R &\triangleright (A \cdot_I B) \\ (K \otimes_P B)_R &\triangleright K_R \otimes_P B_R \\ (K \otimes_P B)_{Q;R} &\triangleright K_Q \otimes_P B_R \end{aligned}$$

**OPT-EXT**

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

**Definition 1.10.4** (sub-register).

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

And we can still 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:

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

*[YX] : For a well-formed quantum register, the sub-register position is well-defined.*

**Definition 1.10.5** (cylinder extension).

$$\begin{aligned} \text{ext}(A, \epsilon) &\equiv A \\ \text{ext}(A, p \ 0) &\equiv A \otimes \mathbf{1} \\ \text{ext}(A, p \ 1) &\equiv \mathbf{1} \otimes A \end{aligned}$$

$$A_{R;R} \triangleright A_R$$

$$\frac{Q \text{ is a subterm of } R \text{ at } p}{A_Q \cdot B_R \triangleright (\text{ext}(A, p) \cdot B)_R}$$

$$\frac{R \text{ is a subterm of } Q \text{ at } p}{A_Q \cdot B_R \triangleright (A \cdot \text{ext}(B, p))_Q}$$

## LABEL-SUM

$$\begin{aligned} (\sum_{i \in S} A)_R &\triangleright \sum_{i \in S} (A_R) \\ (\sum_{i \in S} A)_{Q;R} &\triangleright \sum_{i \in S} (A_{Q;R}) \end{aligned}$$

## LABEL-TEMP

$$\frac{R \text{ is in } Q}{A_R \cdot (B_Q \otimes X) \triangleright (A_R \cdot B_Q) \otimes X}$$

$$\frac{Q \text{ is in } R}{(A_R \otimes X) \cdot B_Q \triangleright (A_R \cdot B_Q) \otimes X}$$

$$A_R \cdot_I (B_R \otimes X) \triangleright (A_R \cdot_I B_R) \cdot X$$

$$(A_R \otimes X) \cdot_I B_R \triangleright (A_R \cdot_I B_Q) \cdot X$$

$$(A \cdot B) \cdot C \triangleright A \cdot (B \cdot C)$$

$$(A + B) \cdot C \triangleright (A \cdot C) + (B \cdot C)$$

$$C \cdot (A + B) \triangleright (C \cdot A) + (C \cdot B)$$

$$\mathbf{1}_O \cdot A \triangleright A$$

$$A \cdot \mathbf{1}_O \triangleright A$$

$$\frac{R \text{ is in } S}{A_{Q;R} \cdot (B_{S;T} \otimes X) \triangleright (A_{Q;R} \cdot B_{S;T}) \otimes X}$$

$$\frac{S \text{ is in } R}{(A_{Q;R} \otimes X) \cdot B_{S;T} \triangleright (A_{Q;R} \cdot B_{S;T}) \otimes X}$$

# Bibliography

- [1] Pablo Arrighi and Gilles Dowek. Linear-algebraic lambda-calculus. 1 2005.
- [2] Pablo Arrighi and Gilles Dowek. Lineal: A linear-algebraic lambda-calculus. *Logical Methods in Computer Science*, Volume 13, Issue 1:1–33, 3 2017.