Qudits: Multi-Level Systems

How to refer to qudits

A collection of qubits are referred to by choosing a symbol, say, **A**.

```
In[ ]:= Let[Qudit, A]
```

The transition operator $|0\rangle\langle 2|$ acting on different qudits.

```
In[ ]:= A[1, 2 → 0]
        A[2, 2 → 0]
        A[3, 2 → 0]
```

Out[ ]= $(|0\rangle\langle 2|)_1$

Out[ ]= $(|0\rangle\langle 2|)_2$

Out[ ]= $(|0\rangle\langle 2|)_3$

In[ ]:= "(" $|0\rangle\langle 2|$ ")"$_3$ // InputForm

Out[ ]//InputForm=

```
A[3, 2 -> 0]
```

```
In[ ]:= A[1, 2, 2 → 0]
        A[2, 2, 2 → 0]
```

Out[ ]= $(|0\rangle\langle 2|)_{1,2}$

Out[ ]= $(|0\rangle\langle 2|)_{2,2}$

Various operators acting on the same qubit.

```
In[ ]:= A[1, 2 → 2]
        A[2, 2 → 1]
        A[3, 2 → 0]
```

Out[ ]= $(|2\rangle\langle 2|)_1$

Out[ ]= $(|1\rangle\langle 2|)_2$

Out[ ]= $(|0\rangle\langle 2|)_3$

In[ ]:=
```
A[1, All] // TableForm
```
Out[ ]//TableForm=

$(|0\rangle\langle 0|)_1$
$(|1\rangle\langle 0|)_1$
$(|2\rangle\langle 0|)_1$
$(|0\rangle\langle 1|)_1$
$(|1\rangle\langle 1|)_1$
$(|2\rangle\langle 1|)_1$
$(|0\rangle\langle 2|)_1$
$(|1\rangle\langle 2|)_1$
$(|2\rangle\langle 2|)_1$

What about more than three levels?

In[ ]:=
```
Let[Qudit, A, Dimension → 4]
```

In[ ]:=
```
A[1, All] // TableForm
```
Out[ ]//TableForm=

$(|0\rangle\langle 0|)_1$
$(|1\rangle\langle 0|)_1$
$(|2\rangle\langle 0|)_1$
$(|3\rangle\langle 0|)_1$
$(|0\rangle\langle 1|)_1$
$(|1\rangle\langle 1|)_1$
$(|2\rangle\langle 1|)_1$
$(|3\rangle\langle 1|)_1$
$(|0\rangle\langle 2|)_1$
$(|1\rangle\langle 2|)_1$
$(|2\rangle\langle 2|)_1$
$(|3\rangle\langle 2|)_1$
$(|0\rangle\langle 3|)_1$
$(|1\rangle\langle 3|)_1$
$(|2\rangle\langle 3|)_1$
$(|3\rangle\langle 3|)_1$

In[ ]:=
```
Dimension[A]
```
Out[ ]=

4

Special flavor index $

The qudit itself is referred to by putting the special flavor index **$** in the last slot of index.

In[◦]:=
```
A[1, $]
A[2, $]
```
Out[◦]=
A_1

Out[◦]=
A_2

In[◦]:=
```
A₂ // InputForm
```
Out[◦]//InputForm=
```
A[2, $]
```

In[◦]:=
```
A[1, 2, $]
A[2, 2, $]
```
Out[◦]=
$A_{1,2}$

Out[◦]=
$A_{2,2}$

Collective reference to several operators on the same qubit

In[◦]:=
```
Let[Qudit, A]
```

In many cases, we need to deal with all transition operators on a particular qudit `A[2,$]`.

In[◦]:=
```
{A[2, 1 → 2], A[2, 2 → 1], A[2, 0 → 1]}
```
Out[◦]=
$\{(|2\rangle\langle 1|)_2, (|1\rangle\langle 2|)_2, (|1\rangle\langle 0|)_2\}$

It can be achieved in a far simpler way.

In[◦]:=
```
A[2, All]
```
Out[◦]=
$\{(|0\rangle\langle 0|)_2, (|1\rangle\langle 0|)_2, (|2\rangle\langle 0|)_2, (|0\rangle\langle 1|)_2, (|1\rangle\langle 1|)_2, (|2\rangle\langle 1|)_2, (|0\rangle\langle 2|)_2, (|1\rangle\langle 2|)_2, (|2\rangle\langle 2|)_2\}$

What about this?

In[◦]:=
```
A[2, {0 → 1, 1 → 2, 2 → 1, 1 → 1}]
```
Out[◦]=
$\{(|1\rangle\langle 0|)_2, (|2\rangle\langle 1|)_2, (|1\rangle\langle 2|)_2, (|1\rangle\langle 1|)_2\}$

Collective reference to many qubits

In[◦]:=
```
Let[Qudit, A]
```

Sometimes, we also need to deal with many qubits.

In[ ]:=
```
A[{1, 2, 3, 4}, $]
```
Out[ ]=

$\{A_1, A_2, A_3, A_4\}$

The same Pauli X operator on many qubits?

In[ ]:=
```
A[{1, 2, 3, 4}, 1 → 1]
```
Out[ ]=

$\{(|1\rangle\langle 1|)_1, (|1\rangle\langle 1|)_2, (|1\rangle\langle 1|)_3, (|1\rangle\langle 1|)_4\}$

Quantum States of Qudits & Operators on Qudits

In[ ]:=
```
Let[Qudit, A]
```

In[ ]:=
```
bs = Basis[A@{1, 2}]
```
Out[ ]=

$\{|0_{A_1}0_{A_2}\rangle, |0_{A_1}1_{A_2}\rangle, |0_{A_1}2_{A_2}\rangle, |1_{A_1}0_{A_2}\rangle, |1_{A_1}1_{A_2}\rangle, |1_{A_1}2_{A_2}\rangle, |2_{A_1}0_{A_2}\rangle, |2_{A_1}1_{A_2}\rangle, |2_{A_1}2_{A_2}\rangle\}$

In[ ]:=
```
vec = Ket[A@{1, 2} → 0] - I * Ket[A@{1, 2} → {2, 1}]
```
Out[ ]=

$|0_{A_1}0_{A_2}\rangle - i\,|2_{A_1}1_{A_2}\rangle$

In[ ]:=
```
op = A[1, 1 → 2] ** A[2, 2 → 0] - 3 * A[1, 0 → 0] + I * A[2, 0 → 2]
```
Out[ ]=

$-3\,(|0\rangle\langle 0|)_1 + i\,(|2\rangle\langle 0|)_2 + (|2\rangle\langle 1|)_1\,(|0\rangle\langle 2|)_2$

In[ ]:=
```
new = op ** vec
```
Out[ ]=

$-3\,|0_{A_1}0_{A_2}\rangle + i\,|0_{A_1}2_{A_2}\rangle$

In[ ]:=
```
col = Matrix[vec];
col // MatrixForm
```
Out[ ]//MatrixForm=

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -i \\ 0 \end{pmatrix}$$

In[•]:=
```
mat = Matrix[op];
mat // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ i & 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & i & 0 & 0 \end{pmatrix}$$

In[•]:=
```
opp = ExpressionFor[mat, A@{1, 2}]
```

Out[•]=

$$-3\,(|0\rangle\langle 0|)_1\,(|0\rangle\langle 0|)_2 + i\,(|0\rangle\langle 0|)_1\,(|2\rangle\langle 0|)_2 - 3\,(|0\rangle\langle 0|)_1\,(|1\rangle\langle 1|)_2 - 3\,(|0\rangle\langle 0|)_1\,(|2\rangle\langle 2|)_2 + i\,(|1\rangle\langle 1|)_1\,(|2\rangle\langle 0|)_2 + (|2\rangle\langle 1|)_1\,(|0\rangle\langle 2|)_2 + i\,(|2\rangle\langle 2|)_1\,(|2\rangle\langle 0|)_2$$

The result is different from the original operator! What's happening here?

The result above equals to the original operator when one takes into account the identity $\hat{I} = \sum_{k=1}^{d} |k\rangle\langle k|$, which `Elaborate` does it for you.

In[•]:=
```
opp - op // Elaborate
```

Out[•]=

0

Summary

Function

- `Let[Qudit,A]`
- `A[1,$]`
- `A[1,1→1], A[1,0→1], A[2,2→1], …`
- `A[1,All], A[1,{1→1,1→0,2→0,…}]`
- `S[{1,2,3,4,...},1→0]`

Related Links

- Tutorial: “Quick Quantum Computing with Q3”