Dyadic Products $|\alpha\rangle\langle\beta|$

Ket vs Bra

In[ ]:=
```
Let[Qubit, S]
```

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

Out[ ]=

$$\{|0_{S_1}0_{S_2}\rangle, |0_{S_1}1_{S_2}\rangle, |1_{S_1}0_{S_2}\rangle, |1_{S_1}1_{S_2}\rangle\}$$

In[ ]:=
```
bb = Dagger[bs]
```

Out[ ]=

$$\{\langle 0_{S_1}0_{S_2}|, \langle 0_{S_1}1_{S_2}|, \langle 1_{S_1}0_{S_2}|, \langle 1_{S_1}1_{S_2}|\}$$

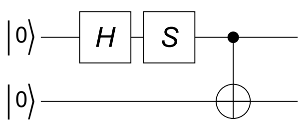
In[ ]:=
```
bs ** Dagger[bs]
```

Out[ ]=

$$\{|0_{S_1}0_{S_2}\rangle\langle 0_{S_1}0_{S_2}|, |0_{S_1}1_{S_2}\rangle\langle 0_{S_1}1_{S_2}|, |1_{S_1}0_{S_2}\rangle\langle 1_{S_1}0_{S_2}|, |1_{S_1}1_{S_2}\rangle\langle 1_{S_1}1_{S_2}|\}$$

In[ ]:=
```
qc = QuantumCircuit[Ket[S@{1, 2}], S[1, 6], S[1, 7], CNOT[S[1], S[2]]]
```

Out[ ]=



In[ ]:=
```
v = Elaborate[qc]
```

Out[ ]=

$$\frac{|0_{S_1}0_{S_2}\rangle}{\sqrt{2}} + \frac{i\,|1_{S_1}1_{S_2}\rangle}{\sqrt{2}}$$

In[ ]:=
```
Dagger[v]
```

Out[ ]=

$$\frac{\langle 0_{S_1}0_{S_2}|}{\sqrt{2}} - \frac{i\,\langle 1_{S_1}1_{S_2}|}{\sqrt{2}}$$

In[ ]:=
```
v ** Dagger[v]
```

Out[ ]=

$$\frac{1}{2}\,|0_{S_1}0_{S_2}\rangle\langle 0_{S_1}0_{S_2}| - \frac{1}{2}\,i\,|0_{S_1}0_{S_2}\rangle\langle 1_{S_1}1_{S_2}| + \frac{1}{2}\,i\,|1_{S_1}1_{S_2}\rangle\langle 0_{S_1}0_{S_2}| + \frac{1}{2}\,|1_{S_1}1_{S_2}\rangle\langle 1_{S_1}1_{S_2}|$$

In[ ]:=
```
w = Ket[S@{1, 2} → 1]
```
Out[ ]=
$\left|1_{S_1}1_{S_2}\right\rangle$

In[ ]:=
```
v ** Dagger[w]
```
Out[ ]=
$$\frac{\left|0_{S_1}0_{S_2}\right\rangle\left\langle 1_{S_1}1_{S_2}\right|}{\sqrt{2}}+\frac{i\left|1_{S_1}1_{S_2}\right\rangle\left\langle 1_{S_1}1_{S_2}\right|}{\sqrt{2}}$$

Ket[...]**Bra[...] might be dangerous!

Let us consider a projection into the one-dimensional subspace spanned by the following vector.

In[ ]:=
```
v = Ket[S[2] → 1] - I * Ket[S[1] → 1] // KetRegulate
```
Out[ ]=
$\left|0_{S_1}1_{S_2}\right\rangle - i\left|1_{S_1}0_{S_2}\right\rangle$

The projection operator is given by the dyadic product. Here, we expect that this projection operator acts non-trivially only on the two qubits `S[1,$]` and `S[2,$]`.

In[ ]:=
```
op = v ** Dagger[v]
```
Out[ ]=
$\left|0_{S_1}1_{S_2}\right\rangle\left\langle 0_{S_1}1_{S_2}\right| + i\left|0_{S_1}1_{S_2}\right\rangle\left\langle 1_{S_1}0_{S_2}\right| - i\left|1_{S_1}0_{S_2}\right\rangle\left\langle 0_{S_1}1_{S_2}\right| + \left|1_{S_1}0_{S_2}\right\rangle\left\langle 1_{S_1}0_{S_2}\right|$

Now, suppose we apply the above dyadic projector on the following vector.

In[ ]:=
```
in = Ket[S@{1, 2, 3} → {0, 1, 1}]
```
Out[ ]=
$\left|0_{S_1}1_{S_2}1_{S_3}\right\rangle$

Unlike our expectation, the projection operator nulls the vector.

In[ ]:=
```
op ** in
```
Out[ ]=
0

This is because of the internal design of Q3 keeping efficiency in mind, and you can see why in this form.

In[ ]:=
```
KetRegulate[op, S@{1, 2, 3}]
```
Out[ ]=
$\left|0_{S_1}1_{S_2}0_{S_3}\right\rangle\left\langle 0_{S_1}1_{S_2}0_{S_3}\right| + i\left|0_{S_1}1_{S_2}0_{S_3}\right\rangle\left\langle 1_{S_1}0_{S_2}0_{S_3}\right| -$
$i\left|1_{S_1}0_{S_2}0_{S_3}\right\rangle\left\langle 0_{S_1}1_{S_2}0_{S_3}\right| + \left|1_{S_1}0_{S_2}0_{S_3}\right\rangle\left\langle 1_{S_1}0_{S_2}0_{S_3}\right|$

Question: how to avoid this? Use `Dyad`.

Dyad

Let us consider a projection into the one-dimensional subspace spanned by the following vector.

In[ ]:=
```
v = Ket[S[2] → 1] - I * Ket[S[1] → 1] // KetRegulate
```
Out[ ]=

$\left|0_{S_1} 1_{S_2}\right\rangle - i \left|1_{S_1} 0_{S_2}\right\rangle$

The projection operator is given by the dyadic product. Here, we expect that this projection operator acts non-trivially only on the two qubits `S[1,$]` and `S[2,$]`.

In[ ]:=
```
op = Dyad[v, v, S@{1, 2}] (* Insted of v**Dagger[v] *)
```
Out[ ]=

$\left|0_{S_1} 1_{S_2}\right\rangle \left\langle 0_{S_1} 1_{S_2}\right| + i \left|0_{S_1} 1_{S_2}\right\rangle \left\langle 1_{S_1} 0_{S_2}\right| - i \left|1_{S_1} 0_{S_2}\right\rangle \left\langle 0_{S_1} 1_{S_2}\right| + \left|1_{S_1} 0_{S_2}\right\rangle \left\langle 1_{S_1} 0_{S_2}\right|$

In[ ]:=
```
op // InputForm
```
Out[ ]//InputForm=
```
Dyad[<|S[1, $] -> 0, S[2, $] -> 1|>, <|S[1, $] -> 0, S[2, $] -> 1|>] +
 I*Dyad[<|S[1, $] -> 0, S[2, $] -> 1|>, <|S[1, $] -> 1, S[2, $] -> 0|>] -
 I*Dyad[<|S[1, $] -> 1, S[2, $] -> 0|>, <|S[1, $] -> 0, S[2, $] -> 1|>] +
 Dyad[<|S[1, $] -> 1, S[2, $] -> 0|>, <|S[1, $] -> 1, S[2, $] -> 0|>]
```

Now, suppose we apply the above dyadic projector on the following vector.

In[ ]:=
```
in = Ket[S@{1, 2, 3} → {0, 1, 1}]
```
Out[ ]=

$\left|0_{S_1} 1_{S_2} 1_{S_3}\right\rangle$

Unlike our expectation, the projection operator nulls the vector.

In[ ]:=
```
op ** in
```
Out[ ]=

$\left|0_{S_1} 1_{S_2} 1_{S_3}\right\rangle - i \left|1_{S_1} 0_{S_2} 1_{S_3}\right\rangle$

Now, as expected, the operator does not affect the third qubit.

How to construct dyadic products?

In[ ]:=
```
v = Ket[S[2] → 1] - I * Ket[S[1] → 1] // KetRegulate
```
Out[ ]=

$\left|0_{S_1} 1_{S_2}\right\rangle - i \left|1_{S_1} 0_{S_2}\right\rangle$

In[ ]:=
```
w = Ket[S@{1, 2} → 1]
```
Out[ ]=

$\left|1_{S_1} 1_{S_2}\right\rangle$

In[ ]:=
```
op = Dyad[v, w, S@{1, 2}]
```
Out[ ]=

$\left|0_{S_1} 1_{S_2}\right\rangle \left\langle 1_{S_1} 1_{S_2}\right| - i \left|1_{S_1} 0_{S_2}\right\rangle \left\langle 1_{S_1} 1_{S_2}\right|$

One can directly type in the specifications such as `Dyad[{…},{…}]`.

In[ ]:=
```
Dyad[{S[1] → 1, S@{1, 2}}, {S[2] → 1, S@{2, 3}}]
```

Out[ ]=

$\left|1_{S_1} 0_{S_2}\right\rangle \left\langle 1_{S_2} 0_{S_3}\right|$

Summary

Functions

- **Dyad**
- **Ket[…]**Bra[…]**

Related Links

- Appendix A of the Quantum Workbook (2022, 2023) -- Available for free via the QuantumPlaybook package.