Numbers as Qubits, Qubit Factorization, Tensor Products and Tensor Powers

by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx

Introduction

This is a tutorial on the use of Quantum`Computing` *Mathematica* add-on to work with binary qubits, tensor products and tensor powers

Load the Package

First load the Quantum`Computing` package. Write:
Needs["Quantum`Computing`"];
then press at the same time the keys SHIFT-ENTER to evaluate. *Mathematica* will load the package:

```
Needs["Quantum`Computing`"]
```

```
Quantum`Computing` Version 2.2.0. (July 2010)
A Mathematica package for Quantum Computing
  in Dirac bra-ket notation and plotting of quantum circuits
by José Luis Gómez-Muñoz

Execute SetComputingAliases[] in order to use
  the keyboard to enter quantum objects in Dirac's notation
SetComputingAliases[] must be executed again in each new notebook that is created
```

In order to use the keyboard to enter quantum objects write:
SetComputingAliases[ ];
then press at the same time the keys SHIFT-ENTER to evaluate. The semicolon prevents *Mathematica* from printing the help message. Remember that SetComputingAliases[ ] must be evaluated again in each new notebook:

```
SetComputingAliases[];
```

From Decimal to Binary Numbers

In order to represent a 9 as a binary number made of 5 qubits, press the keys:
[ESC]toqb[ESC] [TAB] 9 [TAB] 5
then press at the same time the keys SHIFT-ENTER to evaluate:

```
| 9⟩5
```

```
|0_1, 1_2, 0_3, 0_4, 1_5⟩
```

This is the "InputForm" version of the same calculation:

```
DecToQubit[9, 5]
```

```
| 0_1^, 1_2^, 0_3^, 0_4^, 1_5^⟩
```

Copy-paste the previous output as argument of QubitToDec in order to get back the original number:

```
QubitToDec[ | 0_1^, 1_2^, 0_3^, 0_4^, 1_5^⟩]
```

```
9
```

It can be specified which qubit label sould be used for each binary digit:

```
| 9⟩_{10,20,100,200,250}
```

```
| 0_10^, 1_20^, 0_100^, 0_200^, 1_250^⟩
```

Qubit labels that are not integers can be used. Press [ESC]qb[ESC] to obtain the qubit template □^:

```
| 9⟩_{a^,b^,c^,d^,e^}
```

```
| 0_a^, 1_b^, 0_c^, 0_d^, 1_e^⟩
```

This notation can take advantage of standard *Mathematica* commands that generate lists, like Range[]:

```
| 9⟩_Range[12,24,3]
```

```
| 0_12^, 1_15^, 0_18^, 0_21^, 1_24^⟩
```

The "Quantum Register" template [ESC]qr[ESC] works very similar to the standard *Mathematica* command Range, with the extra advantage that we can specify a "prefix" for naming all the qubits:

```
| 9⟩_Register[12,24,3,"w"]
```

```
| 0_w12^, 1_w15^, 0_w18^, 0_w21^, 1_w24^⟩
```

The "Quantum Register" template [ESC]qr[ESC] works very similar to the standard *Mathematica* command Range, with the extra advantage that we can specify a "prefix" for naming all the qubits:

```
| 9⟩_Register[7,"w"]
```

```
| 0_w1^, 0_w2^, 0_w3^, 1_w4^, 0_w5^, 0_w6^, 1_w7^⟩
```

Here we generate a state of ten qubits, with all the qubits equal to zero except the "third" one. Press the keys [ESC]toqb[ESC] to obtain the "decimal to binary" template:

```
| 2^(10-3)⟩_10
```

```
| 0_1̂, 0_2̂, 1_3̂, 0_4̂, 0_5̂, 0_6̂, 0_7̂, 0_8̂, 0_9̂, 0_10̂⟩
```

Here we generate a state of fifteen qubits, with all the qubits equal to zero except the "second" and "fifth" qubits. Press the keys [ESC]toqb[ESC] to obtain the "decimal to binary" template:

```
| 2^(15-2) + 2^(15-5)⟩_15
```

```
| 0_1̂, 1_2̂, 0_3̂, 0_4̂, 1_5̂, 0_6̂, 0_7̂, 0_8̂, 0_9̂, 0_10̂, 0_11̂, 0_12̂, 0_13̂, 0_14̂, 0_15̂⟩
```

Here we generate a state of fifteen qubits, with alternating zeros and ones. Press the keys [ESC]toqb[ESC] to obtain the "decimal to binary" template and [ESC]si[ESC] to obtaing the "sigma notation sum" template:

```
| Sum_(j=1)^(15/2) 2^(15-2 j) ⟩_15
```

```
| 0_1̂, 1_2̂, 0_3̂, 1_4̂, 0_5̂, 1_6̂, 0_7̂, 1_8̂, 0_9̂, 1_10̂, 0_11̂, 1_12̂, 0_13̂, 1_14̂, 0_15̂⟩
```

This is a normalized, equally-weighted, linear combination of the computational basis kets for three qubits:

```
n = 2^3;
1/Sqrt[n] Sum_(j=0)^(n-1) | j⟩_Log[2,n]
```

```
1/(2 Sqrt[2]) ( | 0_1̂, 0_2̂, 0_3̂⟩ + | 0_1̂, 0_2̂, 1_3̂⟩ + | 0_1̂, 1_2̂, 0_3̂⟩ + | 0_1̂,
    1_2̂, 1_3̂⟩ + | 1_1̂, 0_2̂, 0_3̂⟩ + | 1_1̂, 0_2̂, 1_3̂⟩ + | 1_1̂, 1_2̂, 0_3̂⟩ + | 1_1̂, 1_2̂, 1_3̂⟩)
```

This is a factorized version of the ket. It has to be wrapped by Hold, because the factorization is not "stable" under the algebraic rules of this *Mathematica* add-on:

```
n = 2^3;
FactorKet[1/Sqrt[n] Sum_(j=0)^(n-1) | j⟩_Log[2,n]]
```

```
Hold[( | 0_1̂⟩ + | 1_1̂⟩) ⊗ ( | 0_2̂⟩ + | 1_2̂⟩) ⊗ (| 0_3̂⟩ + | 1_3̂⟩)/(2 Sqrt[2])]
```

Copy-paste the output of the previous command **without the Hold[] "wrapper"**, and you obtain an expanded version:

```
(|0_1̂⟩ + |1_1̂⟩) ⊗ (|0_2̂⟩ + |1_2̂⟩) ⊗ (|0_3̂⟩ + |1_3̂⟩)/(2 √2)
```

$$\frac{|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle + |0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle + |0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle + |1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle + |1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle}{2\sqrt{2}}$$

CollectKet gives the answer in a different format: each ket multiplied (or divided) by a scalar:

```
CollectKet[(|0_1̂⟩ + |1_1̂⟩) ⊗ (|0_2̂⟩ + |1_2̂⟩) ⊗ (|0_3̂⟩ + |1_3̂⟩)/(2 √2)]
```

$$\frac{|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}\rangle}{2\sqrt{2}} + \frac{|1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}\rangle}{2\sqrt{2}}$$

More on Qubit Factorization

Kets can be factorized in tensor products using the command **FactorKet**. The result is inside the standard *Mathematica* command **Hold**:

```
FactorKet[|0_1̂, 0_2̂, 0_3̂⟩ + |0_1̂, 0_2̂, 1_3̂⟩]
```

```
Hold[|0_1̂⟩ ⊗ |0_2̂⟩ ⊗ (|0_3̂⟩ + |1_3̂⟩)]
```

The standard *Mathematica* command **ReleaseHold** cancels the **Hold** command, and the factorization evolves to a stable representation:

```
ReleaseHold[Hold[|0_1̂⟩ ⊗ |0_2̂⟩ ⊗ (|0_3̂⟩ + |1_3̂⟩)]]
```

$$|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}\rangle + |0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}\rangle$$

A simple yet nice example of ket factorization:

```
n = 2^4;
FactorKet[Sum_{j=0}^{n-1} 1/√n |j⟩_Log[2,n]]
```

```
Hold[(|0_1̂⟩ + |1_1̂⟩) ⊗ (|0_2̂⟩ + |1_2̂⟩) ⊗ (|0_3̂⟩ + |1_3̂⟩) ⊗ (1/4 (|0_4̂⟩ + |1_4̂⟩))]
```

Copy-paste the output of the previous command **without the Hold[] "wrapper"**, and you obtain a partially expanded version:

```
(|0_1^>+|1_1^>)⊗(|0_2^>+|1_2^>)⊗(|0_3^>+|1_3^>)⊗(1/4 (|0_4^>+|1_4^>))
```

$$\frac{1}{4}\left(|0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}, 0_{\hat{4}}\rangle + |0_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}, 1_{\hat{4}}\rangle\right) + \frac{1}{4}\left(|0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}, 0_{\hat{4}}\rangle + |0_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}, 1_{\hat{4}}\rangle\right) +$$
$$\frac{1}{4}\left(|0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}, 0_{\hat{4}}\rangle + |0_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}, 1_{\hat{4}}\rangle\right) + \frac{1}{4}\left(|0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}, 0_{\hat{4}}\rangle + |0_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}, 1_{\hat{4}}\rangle\right) +$$
$$\frac{1}{4}\left(|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}, 0_{\hat{4}}\rangle + |1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}, 1_{\hat{4}}\rangle\right) + \frac{1}{4}\left(|1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}, 0_{\hat{4}}\rangle + |1_{\hat{1}}, 0_{\hat{2}}, 1_{\hat{3}}, 1_{\hat{4}}\rangle\right) +$$
$$\frac{1}{4}\left(|1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}, 0_{\hat{4}}\rangle + |1_{\hat{1}}, 1_{\hat{2}}, 0_{\hat{3}}, 1_{\hat{4}}\rangle\right) + \frac{1}{4}\left(|1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}, 0_{\hat{4}}\rangle + |1_{\hat{1}}, 1_{\hat{2}}, 1_{\hat{3}}, 1_{\hat{4}}\rangle\right)$$

Use FactorKetList to get a list of the factors:

```
n = 2^4;
FactorKetList[Sum_{j=0}^{n-1} 1/Sqrt[n] |j⟩_Log[2,n]]
```

$$\left\{|0_{\hat{1}}\rangle + |1_{\hat{1}}\rangle,\ |0_{\hat{2}}\rangle + |1_{\hat{2}}\rangle,\ |0_{\hat{3}}\rangle + |1_{\hat{3}}\rangle,\ \frac{1}{4}|0_{\hat{4}}\rangle + \frac{1}{4}|1_{\hat{4}}\rangle\right\}$$

A different example where the "factorization" is actually not very interesting

```
n = 2^4;
FactorKet[Sum_{j=0}^{n-1} 1/Sqrt[n-j] |j⟩_Log[2,n]]
```

$$\text{Hold}\Big[\frac{1}{60\,060}\Big(|0_{\hat{1}}\rangle\otimes|0_{\hat{2}}\rangle\otimes|0_{\hat{3}}\rangle\otimes$$
$$(15\,015\,|0_{\hat{4}}\rangle) + |1_{\hat{1}}\rangle\otimes|0_{\hat{2}}\rangle\otimes|0_{\hat{3}}\rangle\otimes\left(15\,015\sqrt{2}\,|0_{\hat{4}}\rangle\right) + |0_{\hat{1}}\rangle\otimes|1_{\hat{2}}\rangle\otimes|0_{\hat{3}}\rangle\otimes$$
$$\left(10\,010\sqrt{3}\,|0_{\hat{4}}\rangle\right) + |1_{\hat{1}}\rangle\otimes|1_{\hat{2}}\rangle\otimes|0_{\hat{3}}\rangle\otimes(30\,030\,|0_{\hat{4}}\rangle) + |0_{\hat{1}}\rangle\otimes|0_{\hat{2}}\rangle\otimes|1_{\hat{3}}\rangle\otimes$$
$$\left(4290\sqrt{14}\,|0_{\hat{4}}\rangle\right) + |1_{\hat{1}}\rangle\otimes|0_{\hat{2}}\rangle\otimes|1_{\hat{3}}\rangle\otimes\left(10\,010\sqrt{6}\,|0_{\hat{4}}\rangle\right) + |0_{\hat{1}}\rangle\otimes|1_{\hat{2}}\rangle\otimes|1_{\hat{3}}\rangle\otimes$$
$$\left(6006\sqrt{10}\,|0_{\hat{4}}\rangle\right) + |1_{\hat{1}}\rangle\otimes|1_{\hat{2}}\rangle\otimes|1_{\hat{3}}\rangle\otimes\left(30\,030\sqrt{2}\,|0_{\hat{4}}\rangle\right) + |0_{\hat{1}}\rangle\otimes|0_{\hat{2}}\rangle\otimes|0_{\hat{3}}\rangle\otimes$$
$$\left(4004\sqrt{15}\,|1_{\hat{4}}\rangle\right) + |1_{\hat{1}}\rangle\otimes|0_{\hat{2}}\rangle\otimes|0_{\hat{3}}\rangle\otimes\left(8580\sqrt{7}\,|1_{\hat{4}}\rangle\right) + |0_{\hat{1}}\rangle\otimes|1_{\hat{2}}\rangle\otimes|0_{\hat{3}}\rangle\otimes$$
$$\left(5460\sqrt{11}\,|1_{\hat{4}}\rangle\right) + |1_{\hat{1}}\rangle\otimes|1_{\hat{2}}\rangle\otimes|0_{\hat{3}}\rangle\otimes\left(20\,020\sqrt{3}\,|1_{\hat{4}}\rangle\right) + |0_{\hat{1}}\rangle\otimes|0_{\hat{2}}\rangle\otimes|1_{\hat{3}}\rangle\otimes$$
$$\left(4620\sqrt{13}\,|1_{\hat{4}}\rangle\right) + |1_{\hat{1}}\rangle\otimes|0_{\hat{2}}\rangle\otimes|1_{\hat{3}}\rangle\otimes\left(12\,012\sqrt{5}\,|1_{\hat{4}}\rangle\right) + |0_{\hat{1}}\rangle\otimes|1_{\hat{2}}\rangle\otimes|1_{\hat{3}}\rangle\otimes$$
$$(20\,020\,|1_{\hat{4}}\rangle) + |1_{\hat{1}}\rangle\otimes|1_{\hat{2}}\rangle\otimes|1_{\hat{3}}\rangle\otimes(60\,060\,|1_{\hat{4}}\rangle)\Big)\Big]$$

Copy-paste the output of the previous command **without the Hold[] "wrapper"**, and you obtain a partially expanded version:

```
1/60060 ( | 0_1^ ⊗ | 0_2^ ⊗ | 0_3^ ⊗
    (15015 | 0_4^)) + | 1_1^ ⊗ | 0_2^ ⊗ | 0_3^ ⊗ (15015 √2 | 0_4^)) + | 0_1^ ⊗ | 1_2^ ⊗ | 0_3^ ⊗
    (10010 √3 | 0_4^)) + | 1_1^ ⊗ | 1_2^ ⊗ | 0_3^ ⊗ (30030 | 0_4^)) + | 0_1^ ⊗ | 0_2^ ⊗ | 1_3^ ⊗
    (4290 √14 | 0_4^)) + | 1_1^ ⊗ | 0_2^ ⊗ | 1_3^ ⊗ (10010 √6 | 0_4^)) + | 0_1^ ⊗ | 1_2^ ⊗ | 1_3^ ⊗
    (6006 √10 | 0_4^)) + | 1_1^ ⊗ | 1_2^ ⊗ | 1_3^ ⊗ (30030 √2 | 0_4^)) + | 0_1^ ⊗ | 0_2^ ⊗ | 0_3^ ⊗
    (4004 √15 | 1_4^)) + | 1_1^ ⊗ | 0_2^ ⊗ | 0_3^ ⊗ (8580 √7 | 1_4^)) + | 0_1^ ⊗ | 1_2^ ⊗ | 0_3^ ⊗
    (5460 √11 | 1_4^)) + | 1_1^ ⊗ | 1_2^ ⊗ | 0_3^ ⊗ (20020 √3 | 1_4^)) + | 0_1^ ⊗ | 0_2^ ⊗ | 1_3^ ⊗
    (4620 √13 | 1_4^)) + | 1_1^ ⊗ | 0_2^ ⊗ | 1_3^ ⊗ (12012 √5 | 1_4^)) + | 0_1^ ⊗ | 1_2^ ⊗ | 1_3^ ⊗
    (20020 | 1_4^)) + | 1_1^ ⊗ | 1_2^ ⊗ | 1_3^ ⊗ (60060 | 1_4^)))
```

```
1/60060 (15015 | 0_1^, 0_2^, 0_3^, 0_4^⟩ + 4004 √15 | 0_1^, 0_2^, 0_3^, 1_4^⟩ +
    4290 √14 | 0_1^, 0_2^, 1_3^, 0_4^⟩ + 4620 √13 | 0_1^, 0_2^, 1_3^, 1_4^⟩ +
    10010 √3 | 0_1^, 1_2^, 0_3^, 0_4^⟩ + 5460 √11 | 0_1^, 1_2^, 0_3^, 1_4^⟩ + 6006 √10 | 0_1^, 1_2^, 1_3^, 0_4^⟩ +
    20020 | 0_1^, 1_2^, 1_3^, 1_4^⟩ + 15015 √2 | 1_1^, 0_2^, 0_3^, 0_4^⟩ + 8580 √7 | 1_1^, 0_2^, 0_3^, 1_4^⟩ +
    10010 √6 | 1_1^, 0_2^, 1_3^, 0_4^⟩ + 12012 √5 | 1_1^, 0_2^, 1_3^, 1_4^⟩ + 30030 | 1_1^, 1_2^, 0_3^, 0_4^⟩ +
    20020 √3 | 1_1^, 1_2^, 0_3^, 1_4^⟩ + 30030 √2 | 1_1^, 1_2^, 1_3^, 0_4^⟩ + 60060 | 1_1^, 1_2^, 1_3^, 1_4^⟩)
```

CollectKet gives the answer in a different format: each ket multiplied (or divided) by a scalar:

```
CollectKet[
  1/60060 (15015 | 0_1^, 0_2^, 0_3^, 0_4^⟩ + 4004 √15 | 0_1^, 0_2^, 0_3^, 1_4^⟩ + 4290 √14 | 0_1^, 0_2^, 1_3^, 0_4^⟩ +
    4620 √13 | 0_1^, 0_2^, 1_3^, 1_4^⟩ + 10010 √3 | 0_1^, 1_2^, 0_3^, 0_4^⟩ +
    5460 √11 | 0_1^, 1_2^, 0_3^, 1_4^⟩ + 6006 √10 | 0_1^, 1_2^, 1_3^, 0_4^⟩ +
    20020 | 0_1^, 1_2^, 1_3^, 1_4^⟩ + 15015 √2 | 1_1^, 0_2^, 0_3^, 0_4^⟩ + 8580 √7 | 1_1^, 0_2^, 0_3^, 1_4^⟩ +
    10010 √6 | 1_1^, 0_2^, 1_3^, 0_4^⟩ + 12012 √5 | 1_1^, 0_2^, 1_3^, 1_4^⟩ + 30030 | 1_1^, 1_2^, 0_3^, 0_4^⟩ +
    20020 √3 | 1_1^, 1_2^, 0_3^, 1_4^⟩ + 30030 √2 | 1_1^, 1_2^, 1_3^, 0_4^⟩ + 60060 | 1_1^, 1_2^, 1_3^, 1_4^⟩)]
```

```
1/4 | 0_1^, 0_2^, 0_3^, 0_4^⟩ + | 0_1^, 0_2^, 0_3^, 1_4^⟩/√15 + | 0_1^, 0_2^, 1_3^, 0_4^⟩/√14 + | 0_1^, 0_2^, 1_3^, 1_4^⟩/√13 +
  | 0_1^, 1_2^, 0_3^, 0_4^⟩/(2 √3) + | 0_1^, 1_2^, 0_3^, 1_4^⟩/√11 + | 0_1^, 1_2^, 1_3^, 0_4^⟩/√10 + 1/3 | 0_1^, 1_2^, 1_3^, 1_4^⟩ +
  | 1_1^, 0_2^, 0_3^, 0_4^⟩/(2 √2) + | 1_1^, 0_2^, 0_3^, 1_4^⟩/√7 + | 1_1^, 0_2^, 1_3^, 0_4^⟩/√6 + | 1_1^, 0_2^, 1_3^, 1_4^⟩/√5 +
  1/2 | 1_1^, 1_2^, 0_3^, 0_4^⟩ + | 1_1^, 1_2^, 0_3^, 1_4^⟩/√3 + | 1_1^, 1_2^, 1_3^, 0_4^⟩/√2 + | 1_1^, 1_2^, 1_3^, 1_4^⟩
```

Tensor Products

Press the following keyboard keys in order to obtain the template for the "tensor product of a qubit":
[ESC]tprodqb[ESC]
then press [TAB] several times to select the place holder (square) at the bottom of the template and write 1. Press [TAB] one more time to select the uppermost place holder and write 7. Finally press [TAB] one more time to select the place holder in the subscript and write 0. Press at the same time [SHIFT]-[ENTER] to evaluate:

```
⨂_{n=1}^{7} | 0_n̂ ⟩
```

```
| 0_1̂, 0_2̂, 0_3̂, 0_4̂, 0_5̂, 0_6̂, 0_7̂ ⟩
```

The result was the same as this large tensor product of qubits. Each qubit is entered with the keys [ESC]qket0[ESC] and each tensor-product symbol between the kets is entered with the keys [ESC]tp[ESC]:

```
| 0_1̂ ⟩ ⊗ | 0_2̂ ⟩ ⊗ | 0_3̂ ⟩ ⊗ | 0_4̂ ⟩ ⊗ | 0_5̂ ⟩ ⊗ | 0_6̂ ⟩ ⊗ | 0_7̂ ⟩
```

```
| 0_1̂, 0_2̂, 0_3̂, 0_4̂, 0_5̂, 0_6̂, 0_7̂ ⟩
```

The [ESC]tprodqb[ESC] template can be used with qubit values that depend on the subindex:

```
⨂_{n=5}^{11} | ((-1)^n + 1)/2 _n̂ ⟩
```

```
| 0_5̂, 1_6̂, 0_7̂, 1_8̂, 0_9̂, 1_10̂, 0_11̂ ⟩
```

There are many different ways to obtain the same result in *Mathematica*, some of them closer to standard Mathematics notation and some of them closer to programming paradigms, like procedural programming or functional programming. Here the standard *Mathematica* commands Boole and EvenQ are used to generate the same output as in the previous example:

```
⨂_{n=5}^{11} | Boole[EvenQ[n]]_n̂ ⟩
```

```
| 0_5̂, 1_6̂, 0_7̂, 1_8̂, 0_9̂, 1_10̂, 0_11̂ ⟩
```

A useful generalization can be used in the "tensor product": Press the following keyboard keys in order to obtain the template for the "tensor product of a qubit":
[ESC]tprodqb[ESC]
then press [TAB] several times to select the place holder (square) at the bottom of the template and write 1. Press [TAB] one more time to select the uppermost place holder and write {1,3,8,10,20}. Finally press [TAB] one more time to select the place holder in the subscript and write 0. Press at the same time [SHIFT]-[ENTER] to evaluate:

$$\bigotimes_{n=1}^{\{1,3,8,10,20\}} |0_{\hat{n}}\rangle$$

$$|0_{\hat{1}}, 0_{\hat{3}}, 0_{\hat{8}}, 0_{\hat{10}}, 0_{\hat{20}}\rangle$$

This notation can take advantage of standard *Mathematica* commands that generate lists, like Range[]:

$$\bigotimes_{n=1}^{\text{Range}[15,27,3]} |0_{\hat{n}}\rangle$$

$$|0_{\hat{15}}, 0_{\hat{18}}, 0_{\hat{21}}, 0_{\hat{24}}, 0_{\hat{27}}\rangle$$

If a value different form 1 is used in the "initial" value of n for this generalization, then this value is used as an "offset":

$$\bigotimes_{n=5}^{\{1,3,8,10,20\}} |0_{\hat{n}}\rangle$$

$$|0_{\hat{5}}, 0_{\hat{7}}, 0_{\hat{12}}, 0_{\hat{14}}, 0_{\hat{24}}\rangle$$

The tensor product can be used to generate complex products, taking advantage of standard *Mathematica* commands like Mod:

$$\bigotimes_{n=1}^{\{1,3,8,10,20\}} |\text{Mod}[n+1, 2]_{\hat{n}}\rangle$$

$$|0_{\hat{1}}, 0_{\hat{3}}, 1_{\hat{8}}, 1_{\hat{10}}, 1_{\hat{20}}\rangle$$

Here is the tensor product of an operator. This time the **[ESC]tprod[ESC]** template is used instead of [ESC]tprodqb[ESC]. Notice that the qubit labels now are "m", which is the dummy index that was used in the product:

$$\bigotimes_{m=1}^{2} (a00\ |0_{\hat{m}}\rangle \cdot \langle 0_{\hat{m}}| + a01\ |0_{\hat{m}}\rangle \cdot \langle 1_{\hat{m}}| + a10\ |1_{\hat{m}}\rangle \cdot \langle 0_{\hat{m}}| + a11\ |1_{\hat{m}}\rangle \cdot \langle 1_{\hat{m}}|)$$

$$\begin{aligned}
&a00^2\ |0_{\hat{1}}, 0_{\hat{2}}\rangle \cdot \langle 0_{\hat{1}}, 0_{\hat{2}}| + a00\,a10\ |0_{\hat{1}}, 1_{\hat{2}}\rangle \cdot \langle 0_{\hat{1}}, 0_{\hat{2}}| + a00\,a10\ |1_{\hat{1}}, 0_{\hat{2}}\rangle \cdot \langle 0_{\hat{1}}, 0_{\hat{2}}| + \\
&a10^2\ |1_{\hat{1}}, 1_{\hat{2}}\rangle \cdot \langle 0_{\hat{1}}, 0_{\hat{2}}| + a00\,a01\ |0_{\hat{1}}, 0_{\hat{2}}\rangle \cdot \langle 0_{\hat{1}}, 1_{\hat{2}}| + a00\,a11\ |0_{\hat{1}}, 1_{\hat{2}}\rangle \cdot \langle 0_{\hat{1}}, 1_{\hat{2}}| + \\
&a01\,a10\ |1_{\hat{1}}, 0_{\hat{2}}\rangle \cdot \langle 0_{\hat{1}}, 1_{\hat{2}}| + a10\,a11\ |1_{\hat{1}}, 1_{\hat{2}}\rangle \cdot \langle 0_{\hat{1}}, 1_{\hat{2}}| + \\
&a00\,a01\ |0_{\hat{1}}, 0_{\hat{2}}\rangle \cdot \langle 1_{\hat{1}}, 0_{\hat{2}}| + a01\,a10\ |0_{\hat{1}}, 1_{\hat{2}}\rangle \cdot \langle 1_{\hat{1}}, 0_{\hat{2}}| + \\
&a00\,a11\ |1_{\hat{1}}, 0_{\hat{2}}\rangle \cdot \langle 1_{\hat{1}}, 0_{\hat{2}}| + a10\,a11\ |1_{\hat{1}}, 1_{\hat{2}}\rangle \cdot \langle 1_{\hat{1}}, 0_{\hat{2}}| + a01^2\ |0_{\hat{1}}, 0_{\hat{2}}\rangle \cdot \langle 1_{\hat{1}}, 1_{\hat{2}}| + \\
&a01\,a11\ |0_{\hat{1}}, 1_{\hat{2}}\rangle \cdot \langle 1_{\hat{1}}, 1_{\hat{2}}| + a01\,a11\ |1_{\hat{1}}, 0_{\hat{2}}\rangle \cdot \langle 1_{\hat{1}}, 1_{\hat{2}}| + a11^2\ |1_{\hat{1}}, 1_{\hat{2}}\rangle \cdot \langle 1_{\hat{1}}, 1_{\hat{2}}|
\end{aligned}$$

On the other hand, if the qubit is a constant (**this time the subscripts are independent of the product dummy-index m**) you get a different result, because it is now the repeated application of the same operator in the same qubit. Use the [ESC]tprod[ESC] template:

```
⨂_{m=1}^{2} (a00 | 0_î⟩ · ⟨0_î | + a01 | 0_î⟩ · ⟨1_î | + a10 | 1_î⟩ · ⟨0_î | + a11 | 1_î⟩ · ⟨1_î |)
```

```
a00^2 | 0_î⟩ · ⟨0_î | + a01 a10 | 0_î⟩ · ⟨0_î | + a00 a10 | 1_î⟩ · ⟨0_î | + a10 a11 | 1_î⟩ · ⟨0_î | +
 a00 a01 | 0_î⟩ · ⟨1_î | + a01 a11 | 0_î⟩ · ⟨1_î | + a01 a10 | 1_î⟩ · ⟨1_î | + a11^2 | 1_î⟩ · ⟨1_î |
```

In order to enter the tensor product of a quantum expression, press the keys:
[ESC]tprod[ESC]
Then press the [TAB] key one or more times to select the first "place holder" □, and press the keys:
j[TAB]1[TAB]3[TAB]
Then press the keys:
[ESC]cnot[ESC]
Then press the [TAB] key one or more times to select the first "place holder" □, and press the keys:
j [TAB] j+1
then press at the same time the keys SHIFT-ENTER to evaluate:

```
⨂_{j=1}^{3} C^{{ĵ}}[NOT_{ĵ+1}]
```

```
C^{{1̂}}[NOT_{2̂}] · C^{{2̂}}[NOT_{3̂}] · C^{{3̂}}[NOT_{4̂}]
```

Here is the operator in Dirac notation:

```
QuantumEvaluate[⨂_{j=1}^{3} C^{{ĵ}}[NOT_{ĵ+1}]]
```

```
| 0_î, 0_2̂, 0_3̂, 0_4̂⟩ · ⟨0_î, 0_2̂, 0_3̂, 0_4̂ | + | 0_î, 0_2̂, 0_3̂, 1_4̂⟩ · ⟨0_î, 0_2̂, 0_3̂, 1_4̂ | + | 0_î, 0_2̂, 1_3̂, 1_4̂⟩ ·
  ⟨0_î, 0_2̂, 1_3̂, 0_4̂ | + | 0_î, 0_2̂, 1_3̂, 0_4̂⟩ · ⟨0_î, 0_2̂, 1_3̂, 1_4̂ | + | 0_î, 1_2̂, 1_3̂, 0_4̂⟩ ·
  ⟨0_î, 1_2̂, 0_3̂, 0_4̂ | + | 0_î, 1_2̂, 1_3̂, 1_4̂⟩ · ⟨0_î, 1_2̂, 0_3̂, 1_4̂ | + | 0_î, 1_2̂, 0_3̂, 1_4̂⟩ ·
  ⟨0_î, 1_2̂, 1_3̂, 0_4̂ | + | 0_î, 1_2̂, 0_3̂, 0_4̂⟩ · ⟨0_î, 1_2̂, 1_3̂, 1_4̂ | + | 1_î, 1_2̂, 0_3̂, 0_4̂⟩ ·
  ⟨1_î, 0_2̂, 0_3̂, 0_4̂ | + | 1_î, 1_2̂, 0_3̂, 1_4̂⟩ · ⟨1_î, 0_2̂, 0_3̂, 1_4̂ | + | 1_î, 1_2̂, 1_3̂, 1_4̂⟩ ·
  ⟨1_î, 0_2̂, 1_3̂, 0_4̂ | + | 1_î, 1_2̂, 1_3̂, 0_4̂⟩ · ⟨1_î, 0_2̂, 1_3̂, 1_4̂ | + | 1_î, 0_2̂, 1_3̂, 0_4̂⟩ ·
  ⟨1_î, 1_2̂, 0_3̂, 0_4̂ | + | 1_î, 0_2̂, 1_3̂, 1_4̂⟩ · ⟨1_î, 1_2̂, 0_3̂, 1_4̂ | + | 1_î, 0_2̂, 0_3̂, 1_4̂⟩ ·
  ⟨1_î, 1_2̂, 1_3̂, 0_4̂ | + | 1_î, 0_2̂, 0_3̂, 0_4̂⟩ · ⟨1_î, 1_2̂, 1_3̂, 1_4̂ |
```

The TraditionalForm can be easier to read

```
TraditionalForm[QuantumEvaluate[⨂_{j=1}^{3} C^{{ĵ}}[NOT_{ĵ+1}]]]
```

```
|0000⟩⟨0000| + |0001⟩⟨0001| + |0011⟩⟨0010| + |0010⟩⟨0011| + |0110⟩⟨0100| + |0111⟩
   ⟨0101| + |0101⟩⟨0110| + |0100⟩⟨0111| + |1100⟩⟨1000| + |1101⟩⟨1001| + |1111⟩
   ⟨1010| + |1110⟩⟨1011| + |1010⟩⟨1100| + |1011⟩⟨1101| + |1001⟩⟨1110| + |1000⟩⟨1111|
```

The standard *Mathematica* commands TeXForm and TraditionalForm can be used to create an expression that can be copy-pasted to a TEX editor

```
TeXForm[TraditionalForm[QuantumEvaluate[⨂_{j=1}^{3} C^{{j}}[NOT_{j+1}]]]]
```

```
|0000\rangle \langle 0000|+|0001\rangle \langle 0001|+|0011\rangle \langle
   0010|+|0010\rangle \langle 0011|+|0110\rangle \langle 0100|+|0111\rangle \langle
   0101|+|0101\rangle \langle 0110|+|0100\rangle \langle 0111|+|1100\rangle \langle
   1000|+|1101\rangle \langle 1001|+|1111\rangle \langle 1010|+|1110\rangle \langle
   1011|+|1010\rangle \langle 1100|+|1011\rangle \langle 1101|+|1001\rangle \langle
   1110|+|1000\rangle \langle 1111|
```

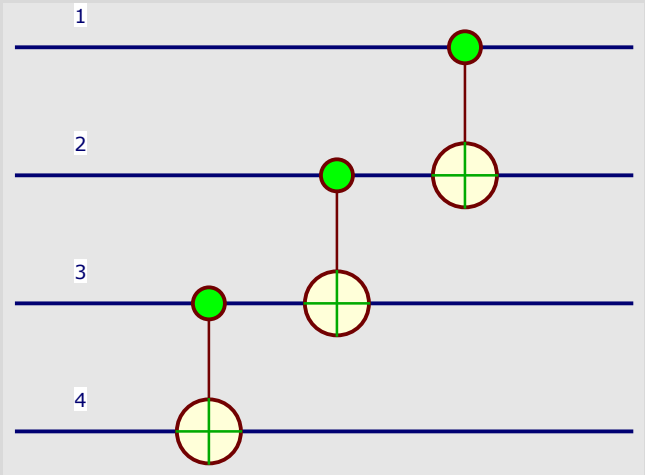
Here is a partial trace of the operator with respect to the second qubit. In order to enter the partial trace template press the keys:
[ESC]trace[ESC]
Then press the [TAB] key and write the qubit, then the [TAB] key and write the operator:

```
QuantumEvaluate[Tr_2[⨂_{j=1}^{3} C^{{j}}[NOT_{j+1}]]]
```

$$\begin{aligned}&|0_1,0_3,0_4\rangle\cdot\langle 0_1,0_3,0_4|+|0_1,1_3,0_4\rangle\cdot\langle 0_1,0_3,0_4|+|0_1,0_3,1_4\rangle\cdot\\&\langle 0_1,0_3,1_4|+|0_1,1_3,1_4\rangle\cdot\langle 0_1,0_3,1_4|+|0_1,0_3,1_4\rangle\cdot\langle 0_1,1_3,0_4|+|0_1,1_3,1_4\rangle\cdot\\&\langle 0_1,1_3,0_4|+|0_1,0_3,0_4\rangle\cdot\langle 0_1,1_3,1_4|+|0_1,1_3,0_4\rangle\cdot\langle 0_1,1_3,1_4|\end{aligned}$$

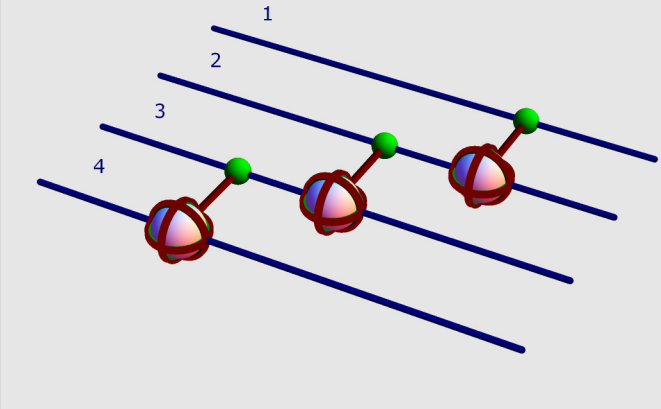
Here is the quantum circuit that represents the gates:

```
QuantumPlot[⨂_{j=1}^{3} C^{{j}}[NOT_{j+1}]]
```



A tridimensitonal representation of the circuit:

```
QuantumPlot3D[⊗_{j=1}^{3} C^{j}[NOT_{j+1}]]
```



Tensor Powers

Press the following keyboard keys in order to obtain the template for the "tensor power of a qubit":
[ESC]tpowqb[ESC]
then press [TAB] several times to select the largest place holder (square) and write 0. Press [TAB] one more time to select the small place holder in the subindex and write 1. Finally press [TAB] one more time to select the place holder in the exponent and write 7. Press at the same time [SHIFT]-[ENTER] to evaluate:

```
(|0_1⟩)^⊗7
```

$$|0_1, 0_2, 0_3, 0_4, 0_5, 0_6, 0_7\rangle$$

If a number different from one is written in the subindex of the tensor power of a qubit, then that number is taken as the starting qubit:

```
(|0_5⟩)^⊗7
```

$$|0_5, 0_6, 0_7, 0_8, 0_9, 0_{10}, 0_{11}\rangle$$

If a list of nonrepeated integers is given in the exponent, those integers are used as channels (qubits). The list of channels (qubits) must have the standard list delimiters of *Mathematica*, the curly brackets {}.

```
(|0_1⟩)^⊗{1,3,8,10,20}
```

$$|0_1, 0_3, 0_8, 0_{10}, 0_{20}\rangle$$

This notation can take advantage of standard *Mathematica* commands that generate lists, like Range[]:

```
(| 0_1^ ⟩)^⊗Range[15,27,3]
```

```
| 0_15^, 0_18^, 0_21^, 0_24^, 0_27^ ⟩
```

If a number different from one is written in the subindex, then a constant is added to all the integers in the list such that qubit 1 becomes the qubit specified in the subindex:

```
(| 0_5^ ⟩)^⊗{1,3,8,10,20}
```

```
| 0_5^, 0_7^, 0_12^, 0_14^, 0_24^ ⟩
```

The "tensor power" of more elaborated expresions can be calculated. It corresponds to have in each qubit the same superposition of 0 and 1, and the total state is the tensor product of all the superpositions.
In order to obtain the template for the "tensor power", press the following keys:
[ESC]tpow[ESC]
then press [TAB] one or two times to select the largest place holder (square) and press:
a[ESC]qket[ESC]+b[ESC]qket[ESC]
then press [TAB] several times to select the first place holder and press
0[TAB]1[TAB]1[TAB]1[TAB]3
Finally press at the same time [SHIFT]-[ENTER] to evaluate.

```
(a | 0_1^ ⟩ + b | 1_1^ ⟩)^⊗3
```

```
a^3 | 0_1^, 0_2^, 0_3^ ⟩ + a^2 b | 0_1^, 0_2^, 1_3^ ⟩ + a^2 b | 0_1^, 1_2^, 0_3^ ⟩ + a b^2 | 0_1^, 1_2^, 1_3^ ⟩ +
 a^2 b | 1_1^, 0_2^, 0_3^ ⟩ + a b^2 | 1_1^, 0_2^, 1_3^ ⟩ + a b^2 | 1_1^, 1_2^, 0_3^ ⟩ + b^3 | 1_1^, 1_2^, 1_3^ ⟩
```

The result was the same as this large product of qubits:

```
Expand[(a | 0_1^ ⟩ + b | 1_1^ ⟩) ⊗ (a | 0_2^ ⟩ + b | 1_2^ ⟩) ⊗ (a | 0_3^ ⟩ + b | 1_3^ ⟩)]
```

```
a^3 | 0_1^, 0_2^, 0_3^ ⟩ + a^2 b | 0_1^, 0_2^, 1_3^ ⟩ + a^2 b | 0_1^, 1_2^, 0_3^ ⟩ + a b^2 | 0_1^, 1_2^, 1_3^ ⟩ +
 a^2 b | 1_1^, 0_2^, 0_3^ ⟩ + a b^2 | 1_1^, 0_2^, 1_3^ ⟩ + a b^2 | 1_1^, 1_2^, 0_3^ ⟩ + b^3 | 1_1^, 1_2^, 1_3^ ⟩
```

A starting qubit and a specific list of qubits can be given:

```
(a | 0_101^ ⟩ + b | 1_101^ ⟩)^⊗{2,6,7}
```

```
a^3 | 0_102^, 0_106^, 0_107^ ⟩ + a^2 b | 0_102^, 0_106^, 1_107^ ⟩ + a^2 b | 0_102^, 1_106^, 0_107^ ⟩ + a b^2 | 0_102^, 1_106^, 1_107^ ⟩ +
 a^2 b | 1_102^, 0_106^, 0_107^ ⟩ + a b^2 | 1_102^, 0_106^, 1_107^ ⟩ + a b^2 | 1_102^, 1_106^, 0_107^ ⟩ + b^3 | 1_102^, 1_106^, 1_107^ ⟩
```

Here is the "tensor power" of an operator. Remember that [ESC]tpow[ESC] has to be used to write the "tensor power" template and [ESC]qketbra[ESC] to write the "one qubit operator element" template.

```
Expand[(a00 | 0_î⟩ · ⟨0_î | + a01 | 0_î⟩ · ⟨1_î | + a10 | 1_î⟩ · ⟨0_î | + a11 | 1_î⟩ · ⟨1_î |)^⊗2]
```

```
a00^2 | 0_î, 0_ĵ⟩ · ⟨0_î, 0_ĵ | + a00 a10 | 0_î, 1_ĵ⟩ · ⟨0_î, 0_ĵ | + a00 a10 | 1_î, 0_ĵ⟩ · ⟨0_î, 0_ĵ | +
 a10^2 | 1_î, 1_ĵ⟩ · ⟨0_î, 0_ĵ | + a00 a01 | 0_î, 0_ĵ⟩ · ⟨0_î, 1_ĵ | + a00 a11 | 0_î, 1_ĵ⟩ · ⟨0_î, 1_ĵ | +
 a01 a10 | 1_î, 0_ĵ⟩ · ⟨0_î, 1_ĵ | + a10 a11 | 1_î, 1_ĵ⟩ · ⟨0_î, 1_ĵ | +
 a00 a01 | 0_î, 0_ĵ⟩ · ⟨1_î, 0_ĵ | + a01 a10 | 0_î, 1_ĵ⟩ · ⟨1_î, 0_ĵ | +
 a00 a11 | 1_î, 0_ĵ⟩ · ⟨1_î, 0_ĵ | + a10 a11 | 1_î, 1_ĵ⟩ · ⟨1_î, 0_ĵ | + a01^2 | 0_î, 0_ĵ⟩ · ⟨1_î, 1_ĵ | +
 a01 a11 | 0_î, 1_ĵ⟩ · ⟨1_î, 1_ĵ | + a01 a11 | 1_î, 0_ĵ⟩ · ⟨1_î, 1_ĵ | + a11^2 | 1_î, 1_ĵ⟩ · ⟨1_î, 1_ĵ |
```

Notice the difference between the previous "tensor power" [ESC]tpow[ESC] and the following "normal power" **[ESC]po[ESC]**:

```
Expand[(a00 | 0_î⟩ · ⟨0_î | + a01 | 0_î⟩ · ⟨1_î | + a10 | 1_î⟩ · ⟨0_î | + a11 | 1_î⟩ · ⟨1_î |)^2]
```

```
a00^2 | 0_î⟩ · ⟨0_î | + a01 a10 | 0_î⟩ · ⟨0_î | + a00 a10 | 1_î⟩ · ⟨0_î | + a10 a11 | 1_î⟩ · ⟨0_î | +
 a00 a01 | 0_î⟩ · ⟨1_î | + a01 a11 | 0_î⟩ · ⟨1_î | + a01 a10 | 1_î⟩ · ⟨1_î | + a11^2 | 1_î⟩ · ⟨1_î |
```

Here is the "tensor power" of a quantum gate. Remember that [ESC]tpow[ESC] has to be used to write the "tensor power" template and [ESC]cnot[ESC] to write the "controlled-not" template.

```
(C^{î}[NOT_ĵ])^⊗3
```

```
C^{î}[NOT_ĵ] · C^{ĵ}[NOT_k̂] · C^{k̂}[NOT_l̂]
```

The tensor power in Dirac notation:

```
QuantumEvaluate[(C^{î}[NOT_ĵ])^⊗3]
```

```
| 0_î, 0_ĵ, 0_k̂, 0_l̂⟩ · ⟨0_î, 0_ĵ, 0_k̂, 0_l̂ | + | 0_î, 0_ĵ, 0_k̂, 1_l̂⟩ · ⟨0_î, 0_ĵ, 0_k̂, 1_l̂ | + | 0_î, 0_ĵ, 1_k̂, 1_l̂⟩ ·
  ⟨0_î, 0_ĵ, 1_k̂, 0_l̂ | + | 0_î, 0_ĵ, 1_k̂, 1_l̂⟩ · ⟨0_î, 0_ĵ, 1_k̂, 1_l̂ | + | 0_î, 1_ĵ, 1_k̂, 0_l̂⟩ ·
  ⟨0_î, 1_ĵ, 0_k̂, 0_l̂ | + | 0_î, 1_ĵ, 0_k̂, 1_l̂⟩ · ⟨0_î, 1_ĵ, 0_k̂, 1_l̂ | + | 0_î, 1_ĵ, 1_k̂, 0_l̂⟩ ·
  ⟨0_î, 1_ĵ, 1_k̂, 0_l̂ | + | 0_î, 1_ĵ, 1_k̂, 1_l̂⟩ · ⟨0_î, 1_ĵ, 1_k̂, 1_l̂ | + | 1_î, 1_ĵ, 0_k̂, 0_l̂⟩ ·
  ⟨1_î, 0_ĵ, 0_k̂, 0_l̂ | + | 1_î, 1_ĵ, 0_k̂, 1_l̂⟩ · ⟨1_î, 0_ĵ, 0_k̂, 1_l̂ | + | 1_î, 1_ĵ, 1_k̂, 1_l̂⟩ ·
  ⟨1_î, 0_ĵ, 1_k̂, 0_l̂ | + | 1_î, 1_ĵ, 1_k̂, 0_l̂⟩ · ⟨1_î, 0_ĵ, 1_k̂, 1_l̂ | + | 1_î, 0_ĵ, 1_k̂, 0_l̂⟩ ·
  ⟨1_î, 1_ĵ, 0_k̂, 0_l̂ | + | 1_î, 0_ĵ, 1_k̂, 1_l̂⟩ · ⟨1_î, 1_ĵ, 0_k̂, 1_l̂ | + | 1_î, 0_ĵ, 0_k̂, 1_l̂⟩ ·
  ⟨1_î, 1_ĵ, 1_k̂, 0_l̂ | + | 1_î, 0_ĵ, 0_k̂, 0_l̂⟩ · ⟨1_î, 1_ĵ, 1_k̂, 1_l̂ |
```

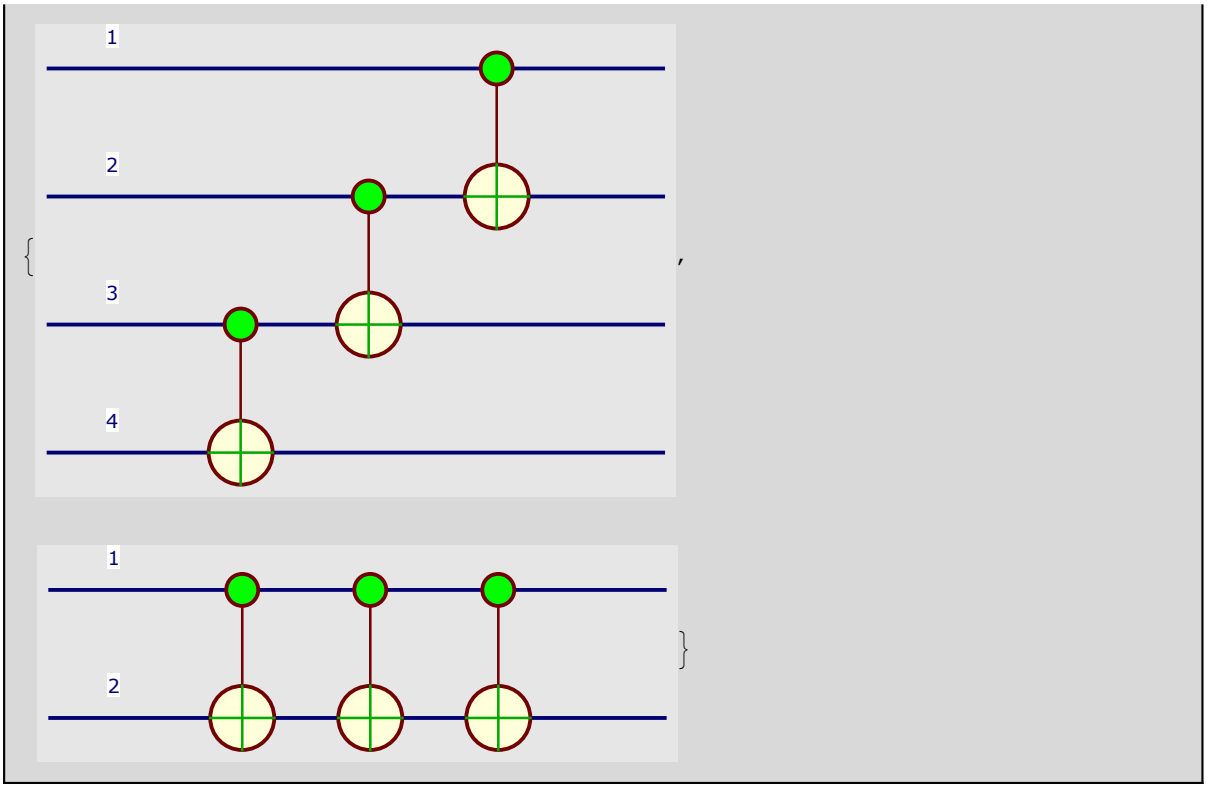
Notice the difference between the previous "tensor power" [ESC]tpow[ESC] and the following "normal power" **[ESC]po[ESC]**:

```
QuantumEvaluate[(C^{î}[NOT[2̂]])^3]
```

```
| 0_î, 0_ĵ⟩ · ⟨0_î, 0_ĵ | + | 0_î, 1_ĵ⟩ · ⟨0_î, 1_ĵ | + | 1_î, 1_ĵ⟩ · ⟨1_î, 0_ĵ | + | 1_î, 0_ĵ⟩ · ⟨1_î, 1_ĵ |
```

Notice the difference between the "**tensor power**" **[ESC]tpow[ESC]** and the "**normal power**" **[ESC]po[ESC]**:

```
{QuantumPlot[(C^{1̂}[NOT_2̂])^⊗3], QuantumPlot[(C^{1̂}[NOT_2̂])^3, QuantumGatePowers → False]}
```



Here is a partial trace of the operator with respect to the second qubit. In order to enter the partial trace template press the keys:
[ESC]trace[ESC]
Then press the [TAB] key and write the qubit, then the [TAB] key and write the operator:

```
QuantumEvaluate[Tr_2̂[(C^{1̂}[NOT_2̂])^⊗3]]
```

$$|0_{\hat{1}}, 0_{\hat{3}}, 0_{\hat{4}}\rangle \cdot \langle 0_{\hat{1}}, 0_{\hat{3}}, 0_{\hat{4}}| + |0_{\hat{1}}, 1_{\hat{3}}, 0_{\hat{4}}\rangle \cdot \langle 0_{\hat{1}}, 0_{\hat{3}}, 0_{\hat{4}}| + |0_{\hat{1}}, 0_{\hat{3}}, 1_{\hat{4}}\rangle \cdot \langle 0_{\hat{1}}, 0_{\hat{3}}, 1_{\hat{4}}| + |0_{\hat{1}}, 1_{\hat{3}}, 1_{\hat{4}}\rangle \cdot \langle 0_{\hat{1}}, 1_{\hat{3}}, 0_{\hat{4}}| + |0_{\hat{1}}, 0_{\hat{3}}, 0_{\hat{4}}\rangle \cdot \langle 0_{\hat{1}}, 1_{\hat{3}}, 1_{\hat{4}}| + |0_{\hat{1}}, 1_{\hat{3}}, 0_{\hat{4}}\rangle \cdot \langle 0_{\hat{1}}, 1_{\hat{3}}, 1_{\hat{4}}|$$

Generalizations of the Tensor Power

Here is a useful generalization of the notation: If a list of 0s and 1s is given in the ket label, then each channel (qubit) takes the corresponding value. The list of 0s and 1s must use the standard list delimiters of *Mathematica*, the curly brackets {}. Remember that [ESC]tpowqb[ESC] has to be used to write the "tensor power of a qubit" template:

```
(|{1, 0, 0, 1}_1̂⟩)^⊗4
```

$$|1_{\hat{1}}, 0_{\hat{2}}, 0_{\hat{3}}, 1_{\hat{4}}\rangle$$

The result was the same as this tensor product of qubits:

```
| 1_î ⟩ ⊗ | 0_ĵ ⟩ ⊗ | 0_k̂ ⟩ ⊗ | 1_l̂ ⟩
```

```
| 1_î, 0_ĵ, 0_k̂, 1_l̂ ⟩
```

A starting qubit and a specific list of qubits can be given:

```
( | {1, 0, 0, 1}_î1 ⟩)^⊗{1,2,6,7}
```

```
| 1_î1, 0_î2, 0_î6, 1_î7 ⟩
```

Even a more general case can be generated if each element in the ket-label list is a pair of constants. Remember that [ESC]tpowqb[ESC] has to be used to write the "tensor power of a qubit" template:

```
( | {{a0, a1}, {b0, b1}}_î ⟩)^⊗2
```

```
a0 b1 | 0_î, 1_ĵ ⟩ + b0 (a0 | 0_î, 0_ĵ ⟩ + a1 | 1_î, 0_ĵ ⟩) + a1 b1 | 1_î, 1_ĵ ⟩
```

The result was the same as this tensor product of qubits:

```
(a0 | 0_î ⟩ + a1 | 1_î ⟩) ⊗ (b0 | 0_ĵ ⟩ + b1 | 1_ĵ ⟩)
```

```
a0 b1 | 0_î, 1_ĵ ⟩ + b0 (a0 | 0_î, 0_ĵ ⟩ + a1 | 1_î, 0_ĵ ⟩) + a1 b1 | 1_î, 1_ĵ ⟩
```

This is a very general example using the [ESC]tpowqb[ESC] template:

```
( | {1, {b0, b1}, 0, {d0, d1}}_k̂0 ⟩)^⊗{3,4,8,9}
```

```
b0 d1 | 1_k̂2, 0_k̂3, 0_k̂7, 1_k̂8 ⟩ +
 d0 (b0 | 1_k̂2, 0_k̂3, 0_k̂7, 0_k̂8 ⟩ + b1 | 1_k̂2, 1_k̂3, 0_k̂7, 0_k̂8 ⟩) + b1 d1 | 1_k̂2, 1_k̂3, 0_k̂7, 1_k̂8 ⟩
```

The result was the same as this tensor product of qubits:

```
| 1_k̂3 ⟩ ⊗ (b0 | 0_k̂4 ⟩ + b1 | 1_k̂4 ⟩) ⊗ | 0_k̂8 ⟩ ⊗ (d0 | 0_k̂9 ⟩ + d1 | 1_k̂9 ⟩)
```

```
b0 d1 | 1_k̂3, 0_k̂4, 0_k̂8, 1_k̂9 ⟩ +
 d0 (b0 | 1_k̂3, 0_k̂4, 0_k̂8, 0_k̂9 ⟩ + b1 | 1_k̂3, 1_k̂4, 0_k̂8, 0_k̂9 ⟩) + b1 d1 | 1_k̂3, 1_k̂4, 0_k̂8, 1_k̂9 ⟩
```

Different Kinds of Products and Powers

Next table shows the different notations for "tensor products", "powers" and "normal powers". Notice the difference betwen a "normal power" a^n and a "tensor power" $(a)^{\otimes n}$. Notice also the difference when the argument of a "tensor product" depends on the dummy index $\otimes_{m=1}^{n} a[m]$ and when it does not depend on the dummy index $\otimes_{m=1}^{n} a$

Expression	Result (in TraditionalForm)
$\text{Expand}\left[\text{QuantumEvaluate}\left[C^{\{\hat{1}\}}\left[\mathcal{NOT}\left[\hat{2}\right]\right]\cdot C^{\{\hat{1}\}}\left[\mathcal{NOT}\left[\hat{2}\right]\right]\right]\right]$	$\lvert 00\rangle\langle 00\rvert + \lvert 01\rangle\langle 01\rvert + \lvert 10\rangle\langle 10\rvert + \lvert 11\rangle\langle 11\rvert$
$\text{Expand}\left[\text{QuantumEvaluate}\left[C^{\{\hat{1}\}}\left[\mathcal{NOT}\left[\hat{2}\right]\right]^2\right]\right]$	$\lvert 00\rangle\langle 00\rvert + \lvert 01\rangle\langle 01\rvert + \lvert 10\rangle\langle 10\rvert + \lvert 11\rangle\langle 11\rvert$
$\text{Expand}\left[\text{QuantumEvaluate}\left[\otimes_{m=1}^{2}\left(C^{\{\hat{1}\}}\left[\mathcal{NOT}\left[\hat{2}\right]\right]\right)\right]\right]$	$\lvert 00\rangle\langle 00\rvert + \lvert 01\rangle\langle 01\rvert + \lvert 10\rangle\langle 10\rvert + \lvert 11\rangle\langle 11\rvert$
$\text{Expand}\left[\text{QuantumEvaluate}\left[C^{\{\hat{1}\}}\left[\mathcal{NOT}\left[\hat{2}\right]\right]\cdot C^{\{\hat{2}\}}\left[\mathcal{NOT}\left[\hat{3}\right]\right]\right]\right]$	$\lvert 000\rangle\langle 000\rvert + \lvert 001\rangle\langle 001\rvert + \lvert 011\rangle\langle 010\rvert + \lvert 010\rangle\langle 011\rvert + \lvert 110\rangle\langle 100\rvert + \lvert 111\rangle\langle 101\rvert + \lvert 101\rangle\langle 110\rvert + \lvert 100\rangle\langle 111\rvert$
$\text{Expand}\left[\text{QuantumEvaluate}\left[\left(C^{\{\hat{1}\}}\left[\mathcal{NOT}\left[\hat{2}\right]\right]\right)^{\otimes_2}\right]\right]$	$\lvert 000\rangle\langle 000\rvert + \lvert 001\rangle\langle 001\rvert + \lvert 011\rangle\langle 010\rvert + \lvert 010\rangle\langle 011\rvert + \lvert 110\rangle\langle 100\rvert + \lvert 111\rangle\langle 101\rvert + \lvert 101\rangle\langle 110\rvert + \lvert 100\rangle\langle 111\rvert$
$\text{Expand}\left[\text{QuantumEvaluate}\left[\otimes_{m=1}^{2}\left(C^{\{\hat{m}\}}\left[\mathcal{NOT}\left[1\hat{+}m\right]\right]\right)\right]\right]$	$\lvert 000\rangle\langle 000\rvert + \lvert 001\rangle\langle 001\rvert + \lvert 011\rangle\langle 010\rvert + \lvert 010\rangle\langle 011\rvert + \lvert 110\rangle\langle 100\rvert + \lvert 111\rangle\langle 101\rvert + \lvert 101\rangle\langle 110\rvert + \lvert 100\rangle\langle 111\rvert$
$\text{Expand}\left[\left(\alpha\ \lvert 1_{\hat{1}}, 1_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 0_{\hat{2}}\rvert + \beta\ \lvert 1_{\hat{1}}, 0_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 1_{\hat{2}}\rvert\right)\otimes\left(\alpha\ \lvert 1_{\hat{1}}, 1_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 0_{\hat{2}}\rvert + \beta\ \lvert 1_{\hat{1}}, 0_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 1_{\hat{2}}\rvert\right)\right]$	$\alpha\,\beta\ \lvert 10\rangle\langle 10\rvert + \alpha\,\beta\ \lvert 11\rangle\langle 11\rvert$
$\text{Expand}\left[\left(\alpha\ \lvert 1_{\hat{1}}, 1_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 0_{\hat{2}}\rvert + \beta\ \lvert 1_{\hat{1}}, 0_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 1_{\hat{2}}\rvert\right)^2\right]$	$\alpha\,\beta\ \lvert 10\rangle\langle 10\rvert + \alpha\,\beta\ \lvert 11\rangle\langle 11\rvert$
$\otimes_{m=1}^{2}\left(\alpha\ \lvert 1_{\hat{1}}, 1_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 0_{\hat{2}}\rvert + \beta\ \lvert 1_{\hat{1}}, 0_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 1_{\hat{2}}\rvert\right)$	$\alpha\,\beta\ \lvert 10\rangle\langle 10\rvert + \alpha\,\beta\ \lvert 11\rangle\langle 11\rvert$
$\text{Expand}\left[\left(\alpha\ \lvert 1_{\hat{1}}, 1_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 0_{\hat{2}}\rvert + \beta\ \lvert 1_{\hat{1}}, 0_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 1_{\hat{2}}\rvert\right)\otimes\left(\alpha\ \lvert 1_{\hat{2}}, 1_{\hat{3}}\rangle\cdot\langle 1_{\hat{2}}, 0_{\hat{3}}\rvert + \beta\ \lvert 1_{\hat{2}}, 0_{\hat{3}}\rangle\cdot\langle 1_{\hat{2}}, 1_{\hat{3}}\rvert\right)\right]$	$\alpha\,\beta\ \lvert 101\rangle\langle 110\rvert + \beta^2\ \lvert 100\rangle\langle 111\rvert$
$\left(\alpha\ \lvert 1_{\hat{1}}, 1_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 0_{\hat{2}}\rvert + \beta\ \lvert 1_{\hat{1}}, 0_{\hat{2}}\rangle\cdot\langle 1_{\hat{1}}, 1_{\hat{2}}\rvert\right)^{\otimes_2}$	$\alpha\,\beta\ \lvert 101\rangle\langle 110\rvert + \beta^2\ \lvert 100\rangle\langle 111\rvert$
$\otimes_{m=1}^{2}\left(\alpha\ \lvert 1_{\hat{m}}, 1_{m\hat{+}1}\rangle\cdot\langle 1_{\hat{m}}, 0_{m\hat{+}1}\rvert + \beta\ \lvert 1_{\hat{m}}, 0_{m\hat{+}1}\rangle\cdot\langle 1_{\hat{m}}, 1_{m\hat{+}1}\rvert\right)$	$\alpha\,\beta\ \lvert 101\rangle\langle 110\rvert + \beta^2\ \lvert 100\rangle\langle 111\rvert$
Expression	Result (in TraditionalForm)

by José Luis Gómez-Muñoz
http://homepage.cem.itesm.mx/lgomez/quantum/
jose.luis.gomez@itesm.mx