Constructing Matrix Kronecker Products

The Wolfram Mathematica command

```
In[1]:= ? KroneckerProduct
```

KroneckerProduct[$m_1, m_2, ...$] constructs the Kronecker product of the arrays m_i . >>

Allows us to easily construct direct products of qubits in order to assemble a multi-qubit register.

We define the matrix representation for the qubit basis vectors $|0\rangle$, $|1\rangle$ in the usual way;

```
In[2]:= ClearAll["Global`*"]
```

zeroket = {1, 0};
oneket = {0, 1};

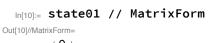
To construct the direct products $|0\rangle \otimes |0\rangle$, $|0\rangle \otimes |1\rangle$, $|1\rangle \otimes |0\rangle$, $|1\rangle \otimes |1\rangle$ we use a combination of the Wolfram KroneckerProduct and Flatten commands, as in

```
In[5]:= state00 = Flatten[KroneckerProduct[zeroket, zeroket]];
    state01 = Flatten[KroneckerProduct[zeroket, oneket]];
    state10 = Flatten[KroneckerProduct[oneket, zeroket]];
    state11 = Flatten[KroneckerProduct[oneket, oneket]];
```



```
In[9]:= state00 // MatrixForm
Out[9]//MatrixForm=

(1
0
0
```





0

```
In[11]:= state10 // MatrixForm
Out[11]//MatrixForm=
\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}
In[12]:= state11 // MatrixForm
Out[12]//MatrixForm=
\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}
```

In agreement with Eqs. (2.55), (2.56) in the text.

Let's define the 3-qubit function

```
In[13]= threeQ[qubit3_, qubit2_, qubit1_] :=
    Flatten[KroneckerProduct[qubit3, qubit2, qubit1]];
```

Exercises

1

(1) Use the function threeQ to construct the eight basis vectors that span a 3-qubit register.

(2) Define a function that inputs 5 qubits and allows construction of their (matrix representation) direct products. Use the latter to find the matrix representation of the state

 $(|00101\rangle - i |10000\rangle + |10101\rangle)/\sqrt{3}$

Operators

We know that the matrix representation of single qubit operators are 2 × 2 square matrices, as for example the Pauli gates, σX , σY , σZ

```
In[14]:= PauliX = \{\{0, 1\}, \{1, 0\}\};
PauliY = \{\{0, -I\}, \{I, 0\}\};
PauliZ = \{\{1, 0\}, \{0, -I\}\};
```

Let's also include the unit operator

```
In[17]:= unit = { { 1, 0 } , { 0, 1 } };
```

We wish to construct the matrix representations of two-qubit operators, e.g. $\sigma X \otimes \sigma Y$. In Mathematica this is accomplished with the *KroneckerProduct* command as in

```
in[18]:= operatorXY = KroneckerProduct[PauliX, PauliY];
```

and which should be expressed as a 2 × 2 matrix; indeed

```
In[19]:= operatorXY // MatrixForm
Out[19]//MatrixForm=
(000-i)
```

```
0 0 i 0
0 - i 0 0
i 0 0 0
```

Lets check if this matrix, when acting on a 2-qubit state, transform the column matrix in a way that is isomorphic to the transformation in ket space. As an example we consider the operation

 $\sigma X \otimes \sigma Y |01\rangle$

Using the definitions introduced in Chapter 1 of the text, the above is equivalent to

 $\sigma X |0\rangle \otimes \sigma Y |1\rangle = |1\rangle \otimes (-i|0\rangle) = -i|10\rangle$

Using the matrix representations,

```
In[20]:= output = operatorXY.state01
```

Out[20]= $\{0, 0, -i, 0\}$

```
In[21]:= output == -I state10
```

Out[21]= True

And so, the matrix representation for $\sigma X \otimes \sigma Y$ acting on the matrix representation of $|01\rangle$ is isomorphic to the result obtained in ket space.

Exercises

(1) Construct the matrix representations of the following 3-qubit operators (here **1** represents the qubit identity operator)

- (a) $\sigma X \otimes \mathbf{1} \otimes \sigma Y$
- (b) $\sigma X \otimes \sigma Y \otimes \mathbf{1}$

(2) Apply the above matrix operators to the matrix vector corresponding to state |000>. Evaluate the result, first using the Dirac ket formalism, and compare with the result obtained with the matrix representation.

(3) Construct the matrix representation for the single-qubit Hadamard gate H. Construct the matrix operator $H \otimes H \otimes H$, and let it operate on state $|000\rangle$. Comment.