An introduction to the Quantum Simulator QCL

Rafael Trapani Possignolo

SEMBEI/LARC/PCS/POLI/USP

August 24, 2011
1. Installation

2. Getting Started
   - Simple examples
   - Simulation Limits
   - QCL Code

3. First examples: Entanglement
   - Measurement
   - Entanglement

4. The first Quantum Operator
   - A simple script
   - Toffoli Gate

5. More on syntax
   - Some helpful structures
Table of Contents

1 Installation

2 Getting Started
   - Simple examples
   - Simulation Limits
   - QCL Code

3 First examples: Entanglement
   - Measurement
   - Entanglement

4 The first Quantum Operator
   - A simple script
   - Toffoli Gate

5 More on syntax
   - Some helpful structures
Get the source code at
http://tph.tuwien.ac.at/~oemer/qcl.html

Untar the file (tar xzvf qcl-0.6.3.tgz)

make; make install

To run, just type qcl in your terminal

Note: qcl is only available in Linux operating systems. If you’re using Windows try a virtual machine or Cygwin.

Note2: You need flex and bison in order to use qcl. Some other packages may also be needed for additional features, look into the documentation to more info.
Table of Contents

1 Installation

2 Getting Started
   • Simple examples
   • Simulation Limits
   • QCL Code

3 First examples: Entanglement
   • Measurement
   • Entanglement

4 The first Quantum Operator
   • A simple script
   • Toffoli Gate

5 More on syntax
   • Some helpful structures
Table of Contents

1 Installation

2 Getting Started
   • Simple examples
   • Simulation Limits
   • QCL Code

3 First examples: Entanglement
   • Measurement
   • Entanglement

4 The first Quantum Operator
   • A simple script
   • Toffoli Gate

5 More on syntax
   • Some helpful structures
Embedded Examples

- The lib folder contain some examples
- Type 'qcl' to start the environment
Embedded Examples

- The lib folder contain some examples
- Type 'qcl' to start the environment
- Type 'include "grover.qcl";' to import the grover.qcl file
- Use 'grover(10);' to run the Grover Search for the number 10
- Try other values
Embedded Examples

- The lib folder contain some examples
- Type ’qcl’ to start the environment
- Type ’include ”grover.qcl”;' to import the grover.qcl file
- Use ’grover(10);’ to run the Grover Search for the number 10
- Try other values
- Now import ’shor.qcl’ and run the Shor’s algorithm to some values
Table of Contents

1 Installation

2 Getting Started
   - Simple examples
   - Simulation Limits
   - QCL Code

3 First examples: Entanglement
   - Measurement
   - Entanglement

4 The first Quantum Operator
   - A simple script
   - Toffoli Gate

5 More on syntax
   - Some helpful structures
Questions

- What happens when you use a prime value?
- What happens when you enter an even number?
- And a square?
Questions

- What happens when you use a prime value?
- What happens when you enter an even number?
- And a square?
- What is the highest factorable value? (Shor)
- What is the highest searchable value? (Grover)
Questions

- What happens when you use a prime value?
- What happens when you enter an even number?
- And a square?
- What is the highest factorable value? (Shor)
- What is the highest searchable value? (Grover)
- What is the maximum number of Qubits allowed? Why?
Table of Contents

1. Installation
2. Getting Started
   - Simple examples
   - Simulation Limits
   - QCL Code
3. First examples: Entanglement
   - Measurement
   - Entanglement
4. The first Quantum Operator
   - A simple script
   - Toffoli Gate
5. More on syntax
   - Some helpful structures
Going through the code

- Use gedit (or your favorite text editor) to open grover.qcl
- Identify the types of 'sub-programs' presented in the language
Getting Started  QCL Code

Going through the code

- Use gedit (or your favorite text editor) to open grover.qcl
- Identify the types of 'sub-programs' presented in the language
- What is the difference between them? (procedure, operator, qufunct)
Going through the code

- Use gedit (or your favorite text editor) to open grover.qcl
- Identify the types of 'sub-programs' presented in the language
- What is the difference between them? (procedure, operator, qufunct)

<table>
<thead>
<tr>
<th>routine type</th>
<th>program state</th>
<th>machine state</th>
<th>recursion</th>
</tr>
</thead>
<tbody>
<tr>
<td>procedure</td>
<td>all</td>
<td>all</td>
<td>yes</td>
</tr>
<tr>
<td>operator</td>
<td>none</td>
<td>unitary</td>
<td>no</td>
</tr>
<tr>
<td>qufunct</td>
<td>none</td>
<td>pseudo-classic</td>
<td>no</td>
</tr>
<tr>
<td>functions</td>
<td>none</td>
<td>none</td>
<td>yes</td>
</tr>
</tbody>
</table>
Going through the code

- Use gedit (or your favorite text editor) to open grover.qcl
- Identify the types of 'sub-programs' presented in the language
- What is the difference between them? (procedure, operator, qufunct)

<table>
<thead>
<tr>
<th>routine type</th>
<th>program state</th>
<th>machine state</th>
<th>recursion</th>
</tr>
</thead>
<tbody>
<tr>
<td>procedure</td>
<td>all</td>
<td>all</td>
<td>yes</td>
</tr>
<tr>
<td>operator</td>
<td>none</td>
<td>unitary</td>
<td>no</td>
</tr>
<tr>
<td>qufunct</td>
<td>none</td>
<td>pseudo-classic</td>
<td>no</td>
</tr>
<tr>
<td>functions</td>
<td>none</td>
<td>none</td>
<td>yes</td>
</tr>
</tbody>
</table>

- Operator $O : A \rightarrow Op \times A$
- qufunc: $F : |x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$
- Can you identify basic quantum ports in the code?
# Table of Contents

1. Installation

2. Getting Started
   - Simple examples
   - Simulation Limits
   - QCL Code

3. First examples: Entanglement
   - Measurement
   - Entanglement

4. The first Quantum Operator
   - A simple script
   - Toffoli Gate

5. More on syntax
   - Some helpful structures
First Examples

- We will make a first example to better understand entanglement and QCL
- First, we will type commands direct on the shell
- The idea is to understand how the simulator works before going to batch mode
Table of Contents

1 Installation

2 Getting Started
   • Simple examples
   • Simulation Limits
   • QCL Code

3 First examples: Entanglement
   • Measurement
   • Entanglement

4 The first Quantum Operator
   • A simple script
   • Toffoli Gate

5 More on syntax
   • Some helpful structures
Open the qcl environment, and type:

- `qureg a[1];` → declare qubit a
- `reset;` → force all qubits to pure zero state
- `measure a;` → make a measure on a

What is the output? Is this deterministic?
Without closing the environment, type:

- reset;
- \( \text{H}(a) \); \quad \rightarrow \quad \text{apply Hadamard transform over a}
- measure a;

What is the output? Is this deterministic? (try re-typing this three commands some times).
Table of Contents

1. Installation
2. Getting Started
   - Simple examples
   - Simulation Limits
   - QCL Code
3. First examples: Entanglement
   - Measurement
   - Entanglement
4. The first Quantum Operator
   - A simple script
   - Toffoli Gate
5. More on syntax
   - Some helpful structures
Re-Open the qcl environment, and type:

- qureg a[1];
- qureg b[1];
- reset;
- H(a);
- CNot(b,a); → flip b if a is $|1\rangle$

What happened? Can you name this state?
Make a final measure on a, what happen to b?
Repeat the commands H, CNot and measure.
Bell states

Without closing the environment, type:

- reset;
- Not(a);
- CNot(b,a); \rightarrow \text{ flip b if a is } |1\rangle

This is the $|\Phi^-\rangle$ Bell-state. Can you produce the missing $|\Psi^+\rangle$ and $|\Psi^-\rangle$ states?
Table of Contents

1 Installation

2 Getting Started
   • Simple examples
   • Simulation Limits
   • QCL Code

3 First examples: Entanglement
   • Measurement
   • Entanglement

4 The first Quantum Operator
   • A simple script
   • Toffoli Gate

5 More on syntax
   • Some helpful structures
# Table of Contents

1. **Installation**
2. **Getting Started**
   - Simple examples
   - Simulation Limits
   - QCL Code
3. **First examples: Entanglement**
   - Measurement
   - Entanglement
4. **The first Quantum Operator**
   - A simple script
   - Toffoli Gate
5. **More on syntax**
   - Some helpful structures
A simple script

We will start by creating a simple script an using it in the QCL environment:

- Create a script.qcl file ('gedit script.qcl &'), containing:

  ```qcl
  include ''shor.qcl'';
  shor(15);
  ```

Now save, open the qcl environment, and call include "teste.qcl".
A simple procedure

Now we will use the procedure feature:

- Create a procedure.qcl file, containing:

  ```qcl
  include ''shor.qcl'';

  procedure callShor(int x){
    shor(x);
  }
  ```

Now save, open the qcl environment, and call `include "procedure.qcl"`. You will have to call `callShor(15)` to run the script.
Table of Contents

1. Installation
2. Getting Started
   - Simple examples
   - Simulation Limits
   - QCL Code
3. First examples: Entanglement
   - Measurement
   - Entanglement
4. The first Quantum Operator
   - A simple script
   - Toffoli Gate
5. More on syntax
   - Some helpful structures
Our first operator

We will use the operator feature to create a Toffoli gate, which will flip a qubit if the other two are 1:

- Create a toffoli.qcl file, containing:

```qcl
operator toffoli(qureg q) {
    if q[0] and q[1] { Not(q[2]); };
}
```
Our first operator

We will use the operator feature to create a Toffoli gate, which will flip a qubit if the other two are 1:

- Create a toffoli.qcl file, containing:

  ```qcl
  operator toffoli(qureg q) {
   if q[0] and q[1] { Not(q[2]); }
  }
  ```

- To test, open qcl and type:

  ```qcl
  include ’’toffoli.qcl’’;
  qureg q[3];
  reset;
  toffoli(q);
  ```
• Result should be 0, as all the qubits are 0. Now, try
  \texttt{Not(q[0]);}
  \texttt{toffoli(q);}
Result should be 0, as all the qubits are 0. Now, try
\[
\text{Not}(q[0]);
\text{toffoli}(q);
\]

Result should be 1. Finally, try
\[
\text{Not}(q[1]);
\text{toffoli}(q);
\]
• Result should be 0, as all the qubits are 0. Now, try
  
  Not(q[0]);
  toffoli(q);

• Result should be 1. Finally, try
  
  Not(q[1]);
  toffoli(q);

• Result should be 7.
# Table of Contents

1. Installation

2. Getting Started
   - Simple examples
   - Simulation Limits
   - QCL Code

3. First examples: Entanglement
   - Measurement
   - Entanglement

4. The first Quantum Operator
   - A simple script
   - Toffoli Gate

5. More on syntax
   - Some helpful structures
## Table of Contents

1. **Installation**
2. **Getting Started**
   - Simple examples
   - Simulation Limits
   - QCL Code
3. **First examples: Entanglement**
   - Measurement
   - Entanglement
4. **The first Quantum Operator**
   - A simple script
   - Toffoli Gate
5. **More on syntax**
   - Some helpful structures
Conditional:

```
if a and (b or c) { do something }
else { do something else };
```

For loop:

```
for i=1 to N { do something }
```

While loop:

```
{ do something } until (some condition);
```

Do and undo transform:

```
H(a); !H(a);
```

Length of qureg a:

```
#a
```
More on syntax

Some helpful structures

More on: http://tph.tuwien.ac.at/~oemer/qcl.html