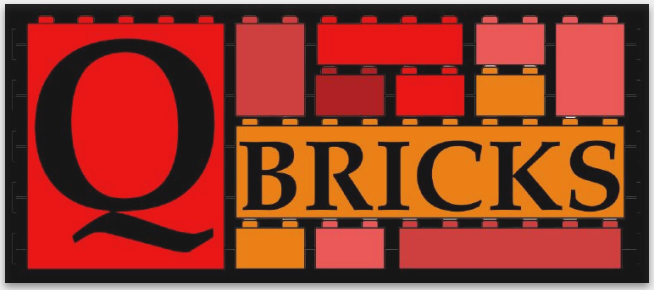
26/06/2024

Graphical output for a quantum circuit description language

Automata and Applications

Author: Benjamin Boyer
Supervisors: Christophe Chareton, Nicolas Nalpon, Quentin Peyras

QBricks project

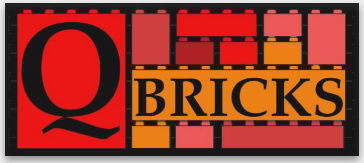


- Open-source environment
- Quantum circuit description language
- Formal proof of quantum programs and their behavior

States | Preconditions | {Instructions} | Postconditions

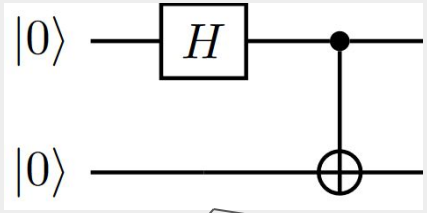
Why visualize circuits?

- Quantum circuit complexity
- Scalability (simplify, prioritize, etc.)
- Interoperability (interfacing with existing tools)
- Interactivity (modify settings, display, etc.)
- Innovation (new fields in evolution)

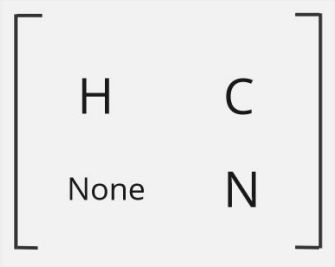


Abstract representation

Bell state preparation circuit

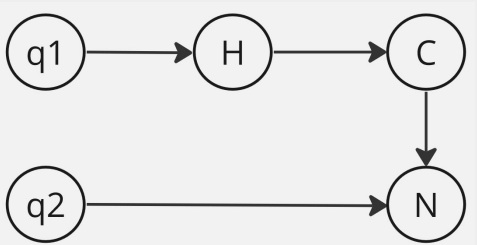


- Circuit composition
- Circuit simplification (rewriting / optimization)
- Customize the visual representation
- Transpiling to other description languages



Matrix

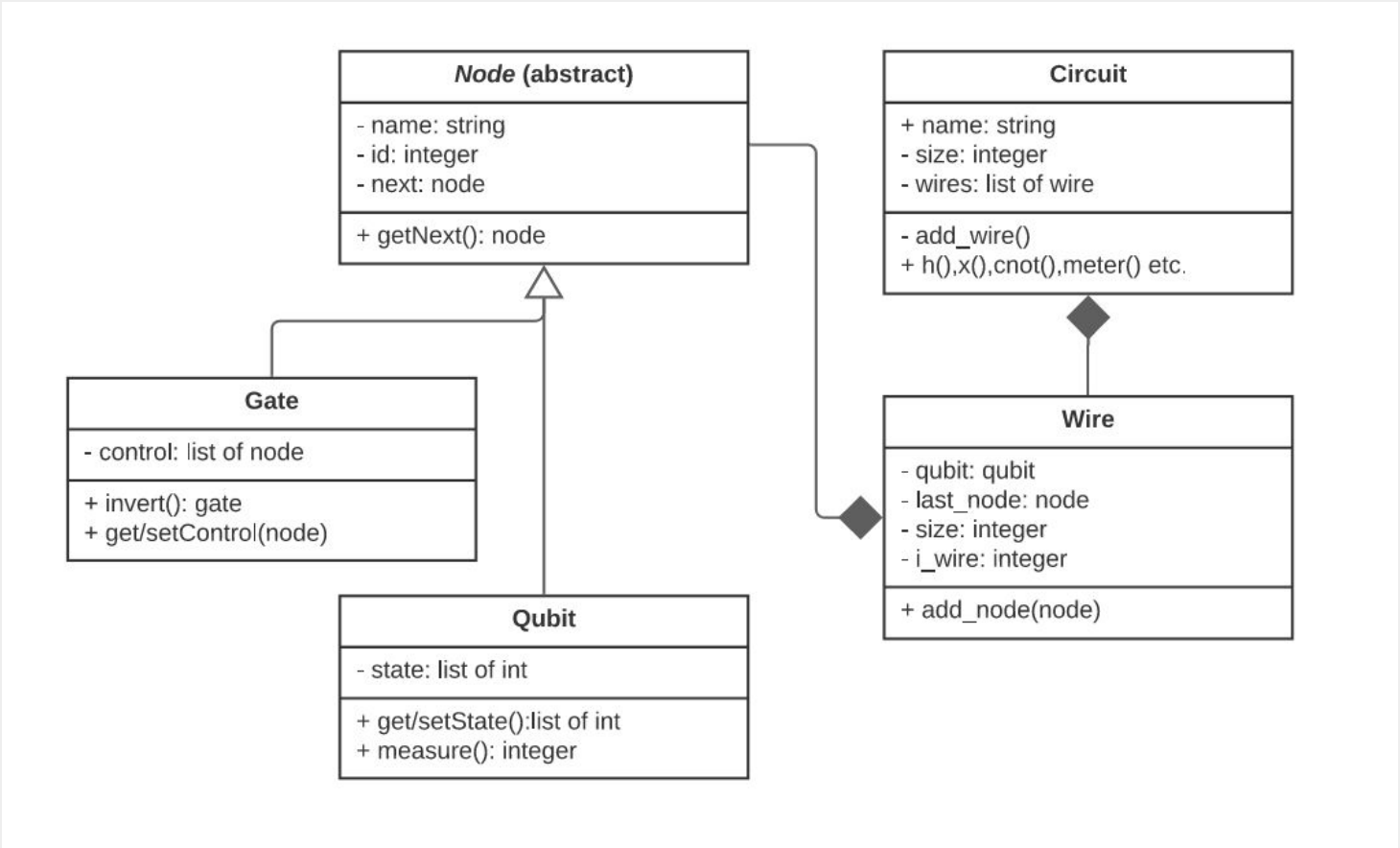
Intuitive and fast
Partial matrix
Costly modification



Graph

Modular
Minimal
Expandable

Class diagram



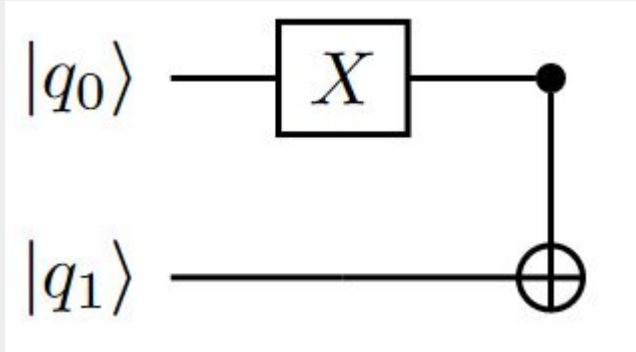
Program example (1)

Create a two-wire circuit

Apply a 'not' gate to wire 0

Apply a 'cnot' gate to wires 0 and 1

```
c0 = Circuit(2)
c0.x(0)
c0.cnot(0, 1)
```




```
print(c0.tikz())
```

Program example (2)

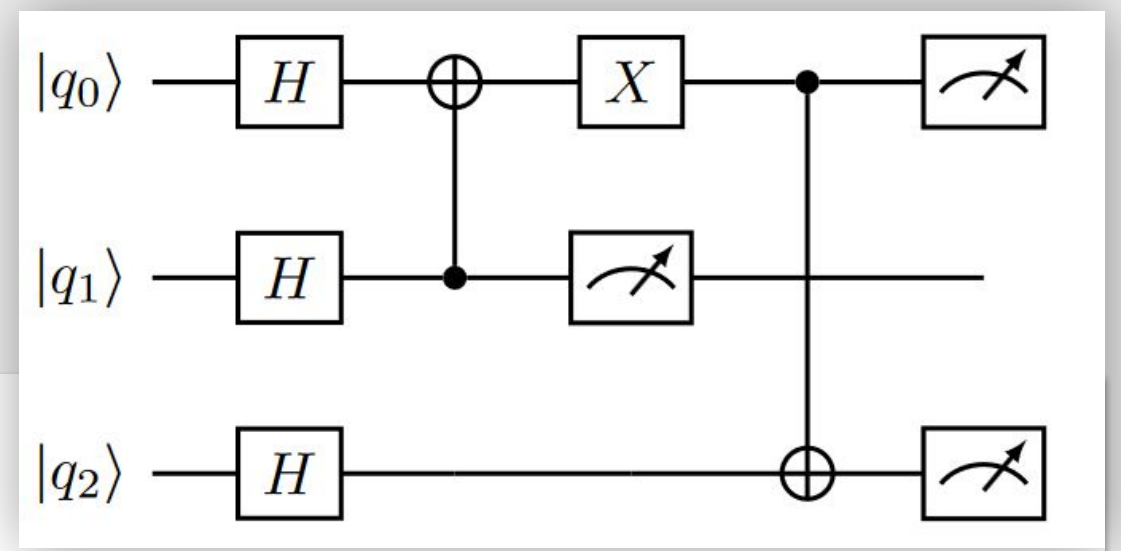
Apply 'c0' sub-circuit to 'main_circuit'

Applies a measurement to all wires in the circuit

Displays generation in Tikz

```
main_circuit = Circuit(3)
main_circuit.h(0, 1, 2)
main_circuit.cnot(1, 0)
main_circuit.func(c0, [0, 2])
main_circuit.meter(all=True)
print(main_circuit.tikz())
```

Tikz rendering

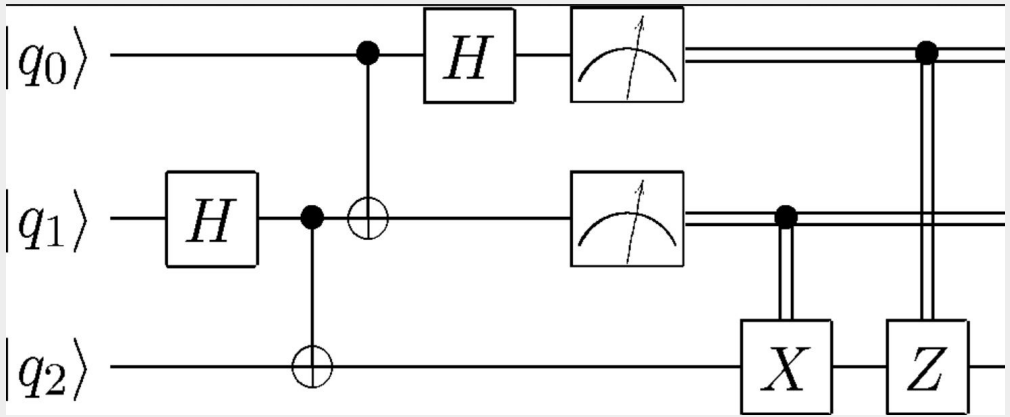



```
\documentclass{standalone}
\usepackage{tikz}
\usetikzlibrary{quantikz}
\begin{document}
\begin{quantikz}[row sep=0.5cm, column sep=0.5cm] \\
\lstick{$\ket{q_0}$} & \gate{H} & \targ{} & \gate{X} & \ctrl{2} & \meter{} \\
\lstick{$\ket{q_1}$} & \gate{H} & \ctrl{-1} & \meter{} & \qw & \qw & \\
\lstick{$\ket{q_2}$} & \gate{H} & \qw & \qw & \targ{} & \meter{} & \\
\end{quantikz}
\end{document}
```

QBricks description language

```
q = NewQreg[3]
b = NewCreg[2]
Had q[1]
Cnot q[1] q[2]
Cnot q[0] q[1]
Had q[0]
b[0] = Measure q[0]
b[1] = Measure q[1]
X q[1] q[2]
Z q[0] q[2]
```

Precise description of a quantum circuit
(its registers, gates, state, etc.)



Quantum teleportation circuit

Interfacing with QBricks

```
q = NewQreg[3]
b = NewCreg[2]
Had q[1]
Cnot q[1] q[2]
Cnot q[0] q[1]
Had q[0]
b[0] = Measure q[0]
b[1] = Measure q[1]
X q[1] q[2]
Z q[0] q[2]
```

Transfers Ast from Ocaml to Python

Abstract representation

Tikz generation (for example)

```
\documentclass{standalone}
\usepackage{tikz}
\usetikzlibrary{quantikz}
\begin{document}
\begin{quantikz}[row sep=0.5cm, column sep=0.5cm]
\lstick{$\ket{q1}$} & \qw & \qw & \gate{H} & \targ{} & \meter{} \\
\lstick{$\ket{q2}$} & \gate{H} & \targ{} & \qw & \ctrl{-1} & \meter{}\\
\lstick{$\ket{q3}$} & \qw & \ctrl{-1} & \qw & \gate{Z}\\
\end{quantikz}
\end{document}
```

In concrete terms

```
q = NewQreg[2]
H q[0]
CNOT q[0] q[1]
```

QBricks source code

In concrete terms

```
q = NewQreg[2]
H q[0]
CNOT q[0] q[1]
```

QBricks source code

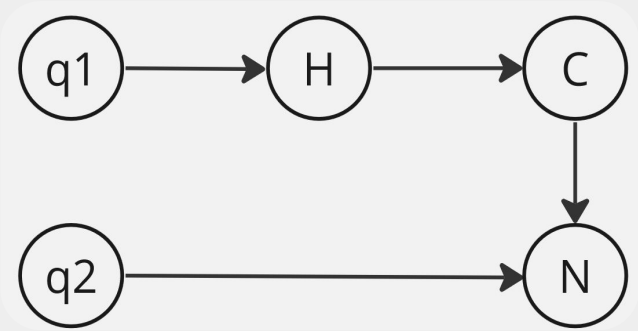
Json AST

In concrete terms

```
q = NewQreg[2]
H q[0]
CNOT q[0] q[1]
```

QBricks source code

Json AST



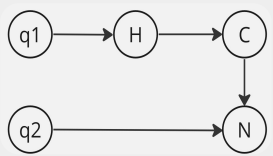
Abstract representation

In concrete terms

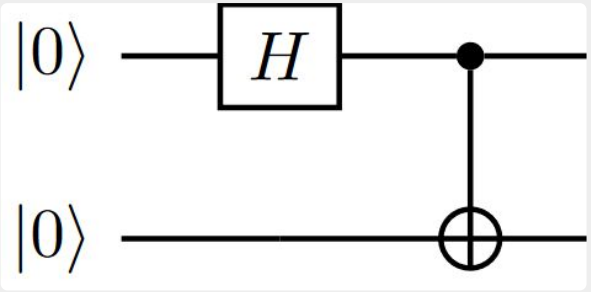
```
q = NewQreg[2]
H q[0]
CNOT q[0] q[1]
```

QBricks
source code

Json AST



Abstract representation



Visual rendering
(for example)

Summary

Done

- Learn the basics of QBricks and Tikz theory and language
- Implement the matrix model
- Implementing the graph model
- Create, manipulate and compile a circuit in Latex
- Creating and using macros in circuits

To do

- Deepening theory
- Interfacing the tool with the QBricks project
- Continue graph implementation
- Expand the range of possible circuits (add gates, loops, etc.).
- Event-driven version
- Other output formats

Conclusion and references

- *Presentation at IQFA Nov 2021 | 12th Colloquium on Quantum Engineering, Fundamental Aspects to Applications*
- *Qbricks at ESOP March 2021 | ESOP conference*
- *https://qiskit.org/documentation/stable/0.24/tutorials/circuits/index.html*
- *“Quantum, un peu de mathématiques pour l’informatique quantique” de Arnaud Bodin*
- *https://www.utc.fr/~wschon/sr06/demonstrateur-algorithmes-quantiques-master/website/_site/porte.html*
- *https://learn.microsoft.com/fr-fr/azure/quantum/concepts-circuits*

Author: Benjamin Boyer
Supervisors: Christophe Chareton, Nicolas Nalpon, Quentin Peyras