Introduction to the Quantum Computations and the Quantum Circuits Model

Michael Kirkedal Thomsen

Department of Computer Science
University of Copenhagen

m.kirkedal@di.ku.dk

Nov 22 2017

Outline

Recap

Recap

Quantum Circuit Model

Performing Quantum Computations

Conclusion

Remember logic circuits

Recap

How do we define and use logic circuits?

Remember logic circuits

Recap

How do we define and use logic circuits?

- AND, OR, NOR, XOR, NOT, ...
- Semantics based on truth tables
- Written as diagrams
- Implemented with e.g. CMOS

A	B	$A \wedge B$	$A \vee B$	$A \oplus B$	$\neg A$
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Why talk about logic circuits

Recap

Why do we have a need to talk about the logic circuit model?

Hint: I have pushed it from the beginning of the course.

Why talk about logic circuits

Recap

Why do we have a need to talk about the logic circuit model?

Hint: I have pushed it from the beginning of the course.

- It is impossible to implement any interesting computer directly in any physical environment.
- Abstractions are important and required to implement computers.

Why talk about logic circuits

Recap

Why do we have a need to talk about the logic circuit model?

Hint: I have pushed it from the beginning of the course.

- It is impossible to implement any interesting computer directly in any physical environment.
- Abstractions are important and required to implement computers.

This also holds true for quantum computers!

Quantum Circuit Model

Recap

Quantum Circuit Model

Performing Quantum Computations

Conclusion

Making a Model for Quantum Computations

Quantum Circuit Model

We know from Anders Sørensen that operations in a quantum computer needs to be reversible.

First step, let's make a reversible logic circuit model based on the standard.

General Properties of Reversible Circuits

Quantum Circuit Model

Most conventional logic gates are irreversible (e.g. AND, OR)

General Properties of Reversible Circuits

Quantum Circuit Model

Most conventional logic gates are irreversible (e.g. AND, OR)

An n-bit gate is *reversible* if

- the number of input lines is equal to the number of output lines ($n \times n$ gates), and
- the logical function $\mathbb{B}^n \to \mathbb{B}^n$ of the gate is bijective.

General Properties of Reversible Circuits

Quantum Circuit Model

Most conventional logic gates are irreversible (e.g. AND, OR)

An n-bit gate is *reversible* if

- the number of input lines is equal to the number of output lines ($n \times n$ gates), and
- the logical function $\mathbb{B}^n \to \mathbb{B}^n$ of the gate is bijective.

A circuit is *reversible* if

- consists only of reversible gates,
- it is combinatorial (acyclic), and
- it does *not* contain fan-out.

Basic Reversible Gates

Quantum Circuit Model

Based on Feynman's diagram notation and Fredkin, Toffoli's model.

[Toffoli, 1980, Fredkin and Toffoli, 1982, Feynman, 1985]

Not gate

A	$\neg A$
0	1
1	0

Basic Reversible Gates

Quantum Circuit Model

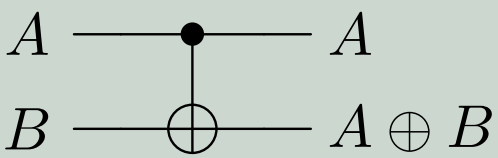
Based on Feynman's diagram notation and Fredkin, Toffoli's model.

[Toffoli, 1980, Fredkin and Toffoli, 1982, Feynman, 1985]

Not gate

A	$\neg A$
0	1
1	0

Controlled-not gate



A	B	A	$A \oplus B$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Basic Reversible Gates

Quantum Circuit Model

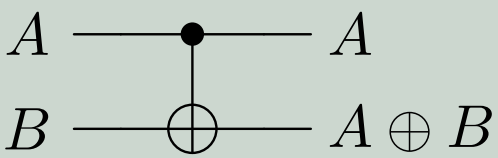
Based on Feynman's diagram notation and Fredkin, Toffoli's model.

[Toffoli, 1980, Fredkin and Toffoli, 1982, Feynman, 1985]

Not gate

A	$\neg A$
0	1
1	0

Controlled-not gate



A	B	A	$A \oplus B$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

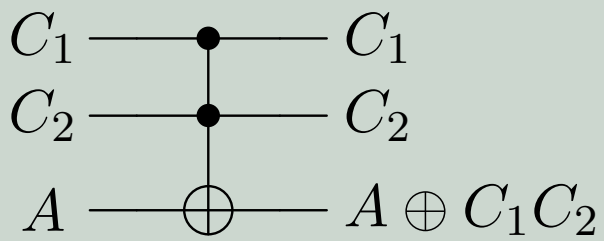
Note, we can also define this as permutation matrices.

Generalising Reversible Gates

Quantum Circuit Model

We can generalise this to more advanced gates.

CC-not gate

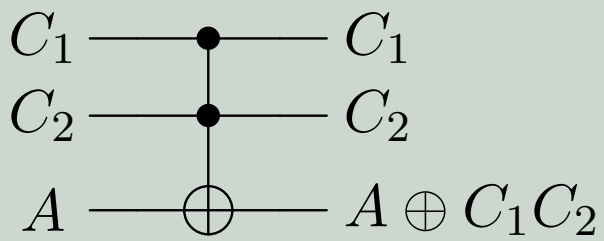


Generalising Reversible Gates

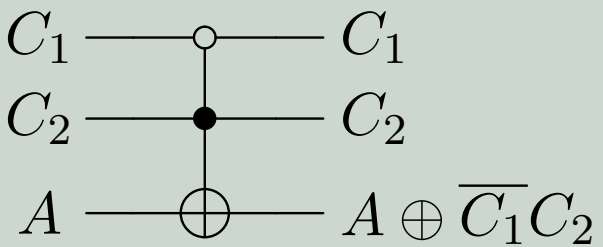
Quantum Circuit Model

We can generalise this to more advanced gates.

CC-not gate



Negative controls

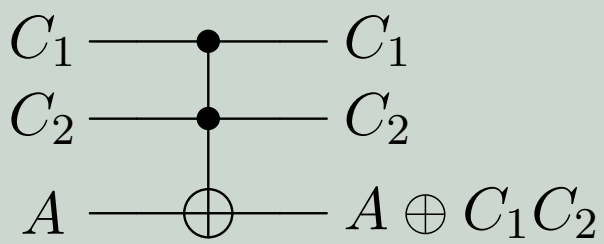


Generalising Reversible Gates

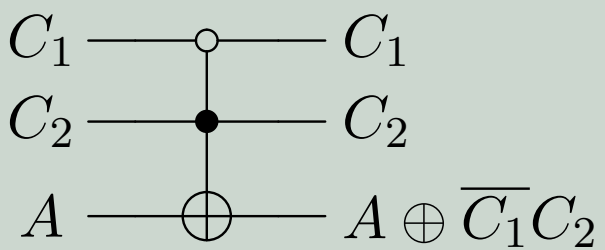
Quantum Circuit Model

We can generalise this to more advanced gates.

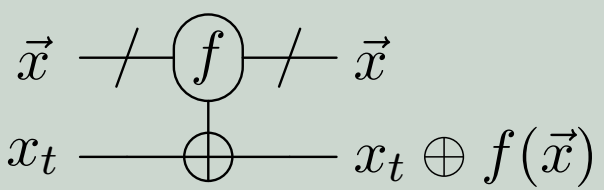
CC-not gate



Negative controls



Control function

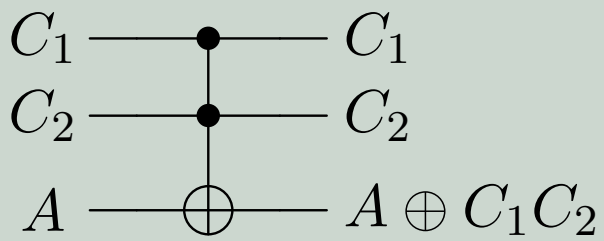


Generalising Reversible Gates

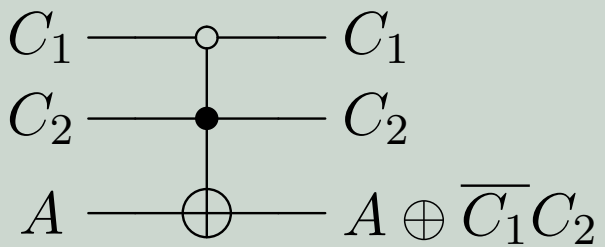
Quantum Circuit Model

We can generalise this to more advanced gates.

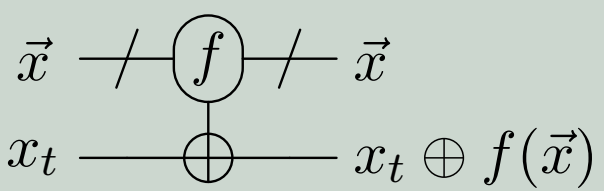
CC-not gate



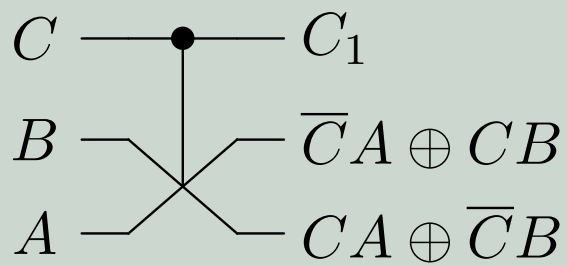
Negative controls



Control function



Controlled-swap gate

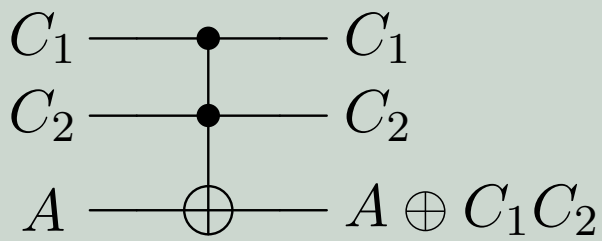


Generalising Reversible Gates

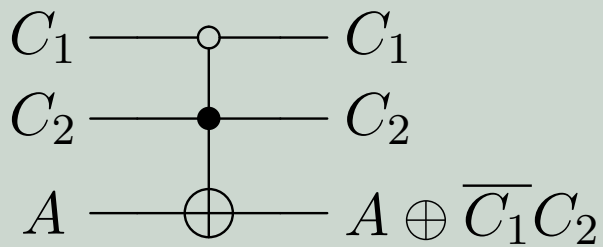
Quantum Circuit Model

We can generalise this to more advanced gates.

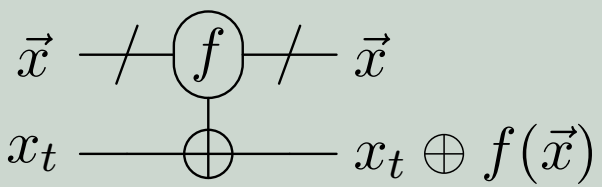
CC-not gate



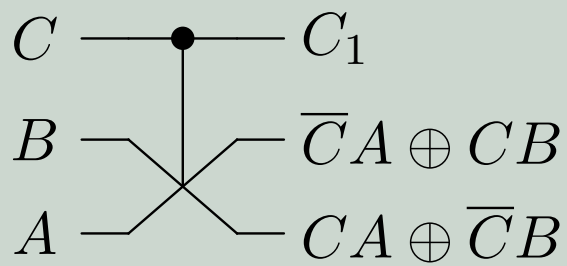
Negative controls



Control function



Controlled-swap gate



Note, these gates are not implementable in current quantum computers.

Second abstraction on Quantum gates

Quantum Circuit Model

What is the consequence of having gates that is not implementable and is it a problem?

Second abstraction on Quantum gates

Quantum Circuit Model

What is the consequence of having gates that is not implementable and is it a problem?

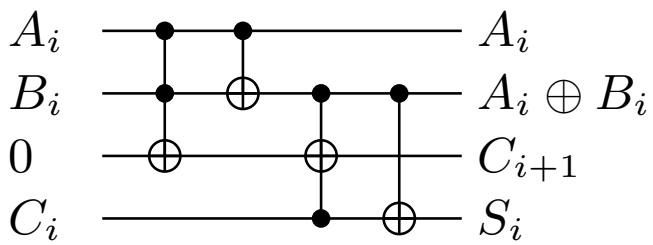
- No, it is not a problem.
  - We already to is for logical circuits; you did it in A2.
  - You do in every time you program.
- We must define translation to the lower-level abstraction; call logic synthesis.
- We can define quantum circuits that is much larger than is executable on current quantum computers.
- As computer scientists we can play with what is possible.

Full-adder approach

Quantum Circuit Model

Adder calculation

$$\begin{aligned} S_i &= C_i \oplus A_i \oplus B_i \\ C_{i+1} &= C_i(A_i \oplus B_i) \oplus A_i B_i \,. \end{aligned}$$

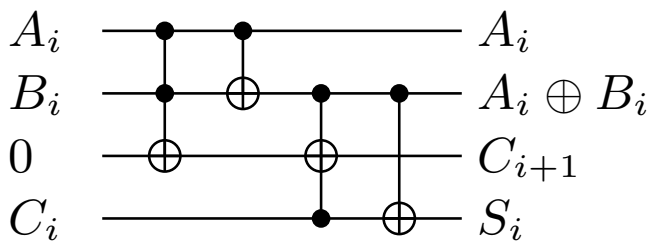


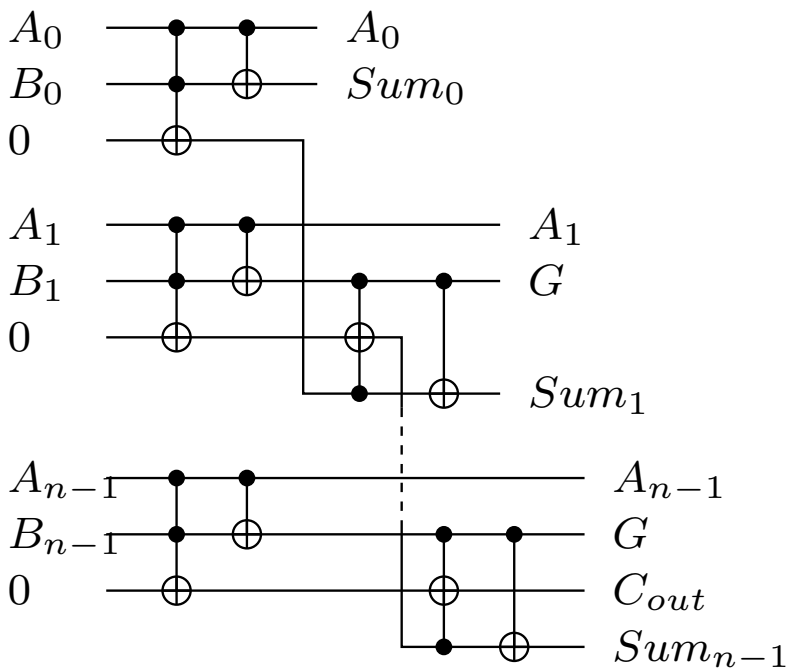
Full-adder approach

Quantum Circuit Model

Adder calculation

$$S_i = C_i \oplus A_i \oplus B_i$$
$$C_{i+1} = C_i(A_i \oplus B_i) \oplus A_i B_i \,.$$



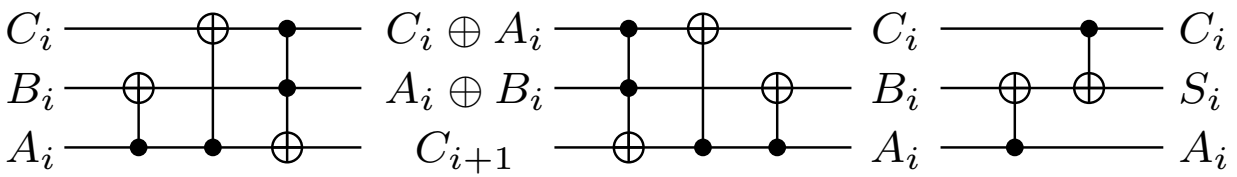


Garbage

Garbage are non-constant output that is not part of the desired embedding.

V-shaped adder

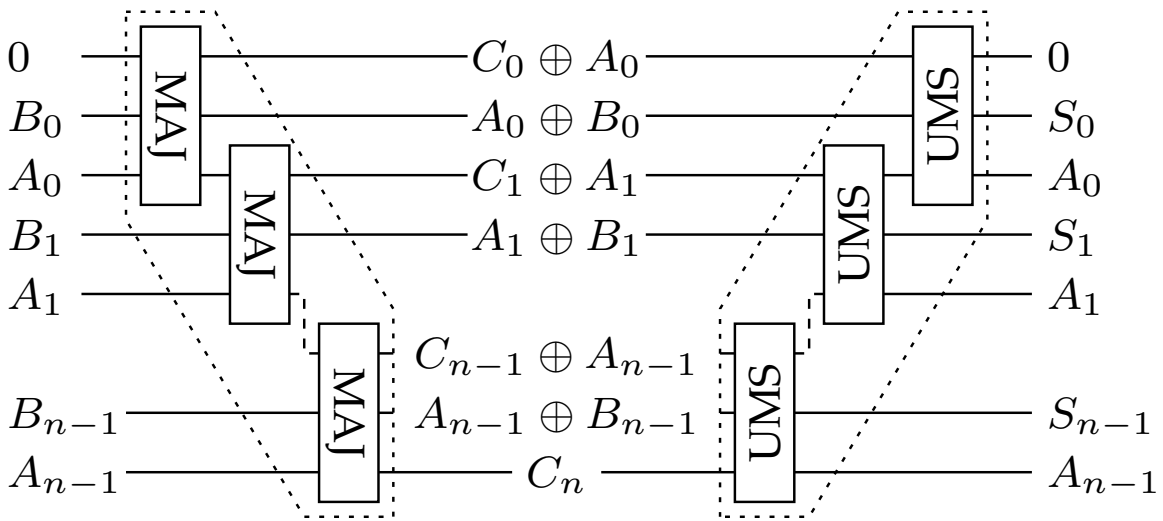
Quantum Circuit Model



Majority Circuit | Unmajority Circuit | Sum calculation Circuit

Insight

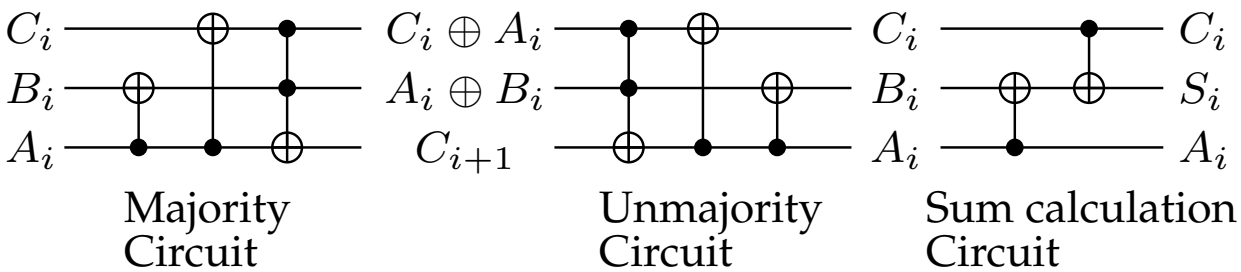
Sum and carry-out independent.



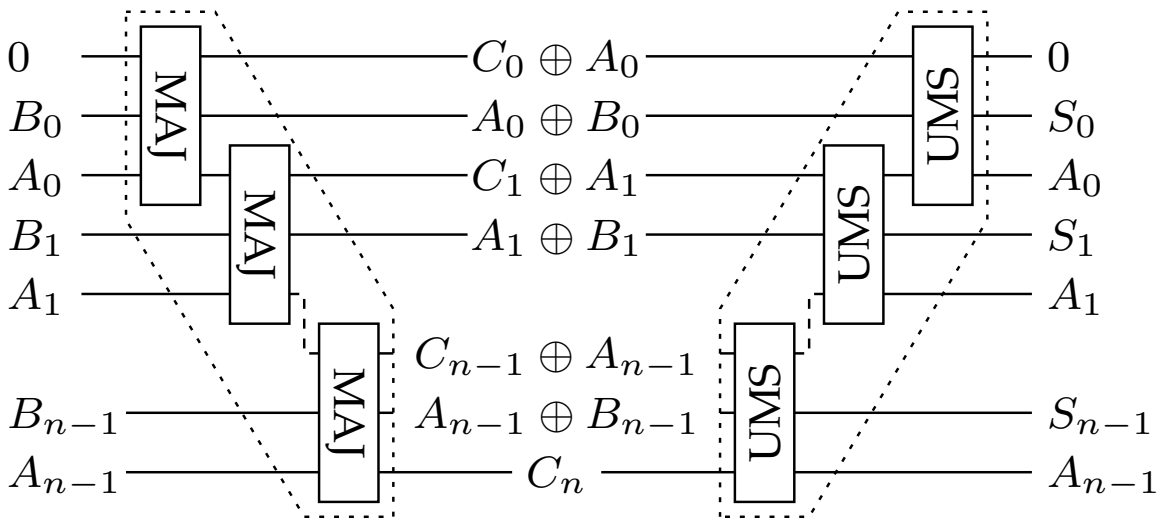
[Vedral et al., 1996, Cuccaro et al., 2005]

V-shaped adder

Quantum Circuit Model



Majority Circuit | Unmajority Circuit | Sum calculation Circuit



Insight

Sum and carry-out independent.

Ancillae

Wires that are constant at both input and output.

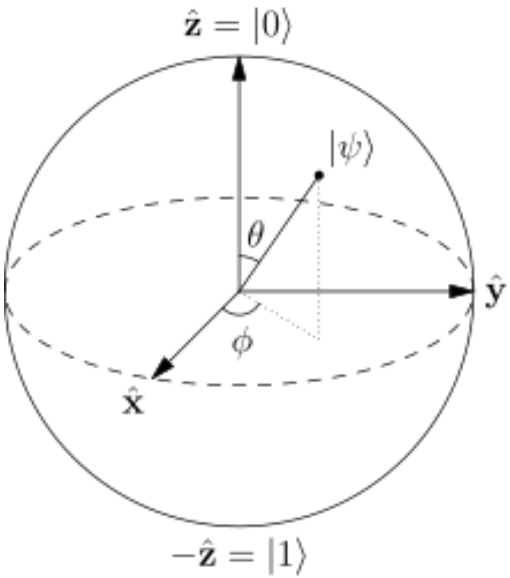
[Vedral et al., 1996, Cuccaro et al., 2005]

The infamous Qubit

Quantum Circuit Model

Reversibility is only half the story.

A qubit can be modelled as a unit vector of the Bloch sphere.



- Usually, the vectors $|0\rangle$ and $|1\rangle$ are used as the standard basis; spin-up and spin-down that Anders told about.
- A value is a linear combination (superposition) of the basis vectors, $\alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$.
- A qubit is thus considered a two-element complex vector $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$.
- Can also be considered as a rotation around two axis.

More Qubits

Quantum Circuit Model

A state of more qubits is then modelled as a product of all standard bases.

- E.g. a two qubit state is
  - $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \theta|11\rangle$, where $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\theta|^2 = 1$.
  - Thus a four-element complex vector.
- A three qubit state is then modelled by a eighth-element complex-valued vector.
- ...

Some Quantum Operations

Quantum Circuit Model

Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Maps the basis states such that a measurement will have equal probabilities to become 1 or 0 (i.e. creates a superposition).

Some Quantum Operations

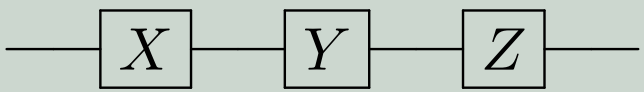
Quantum Circuit Model

Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Maps the basis states such that a measurement will have equal probabilities to become 1 or 0 (i.e. creates a superposition).

Pauli gates



$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Rotations around the axis of the Bloch sphere.

More Quantum Operations

Quantum Circuit Model

T gate

$$T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

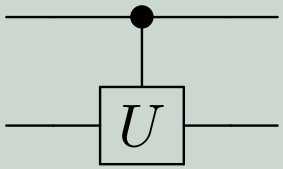
More Quantum Operations

Quantum Circuit Model

T gate

$$T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

Controlled gate



$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix} \qquad \mathrm{C}(U) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}$$

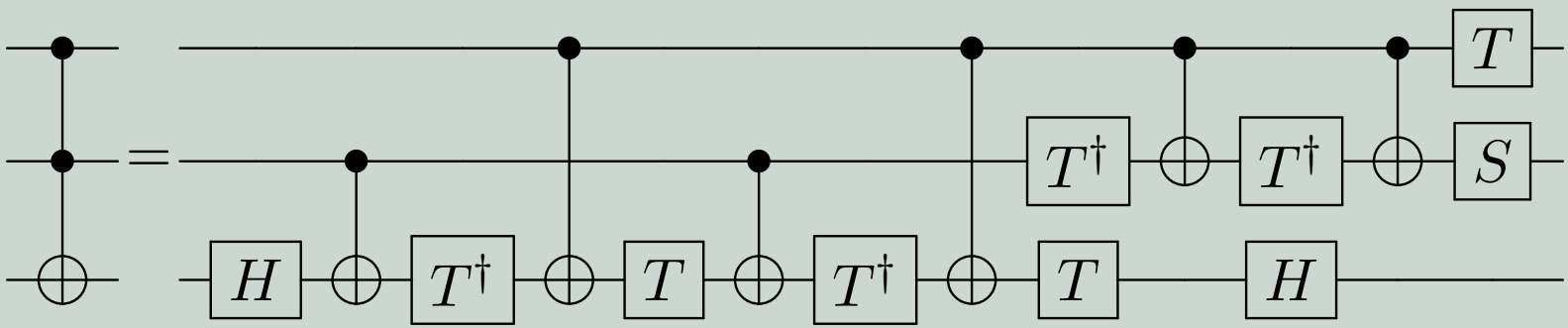
The available gate (and cost of these) differs is different implementations of quantum computers.

Quantum circuit

Quantum Circuit Model

With the smaller gates we can define larger circuits:

CC-not gate



- The task is the find the optimal circuit wrt. some cost metric, which is much dependent on the implementation technology.

Superposition, Entanglement, Measurements

Quantum Circuit Model

Three principles very simplified

- Superposition
  - A qubit is on a superposition when it is in a state that is a combination of the basis states.
- Entanglement
  - Qubits are entangled when the state of one is dependent on the state other.
  - Entanglement is created by an interaction by a qubit in superposition.
- Measurement
  - Projects the state of a qubit onto one of the basis vectors.
  - Operation “destroys” any superposition of a qubit.

Performing Quantum Computations

Recap

Quantum Circuit Model

Performing Quantum Computations

Conclusion

IBM Q

Performing Quantum Computations

- `https://www.research.ibm.com/ibm-q/`
- Online experimental quantum computers

Languages for describing quantum circuits

Performing Quantum Computations

There exist several higher-level languages that can be used to describe quantum circuits

- QASM used in IBM Q
  - Relative low-level language that resembles netlists.
- Quipper
  - DSL embedded in Haskell. Includes the basic gates, but also monadic combinators (map, etc.) that makes it easier to create large circuits
- QWIRE
  - Similar to Quipper, but embedded in Coq.
- Liqui$|\rangle$
  - Stand-alone DSL from Microsoft Research. Integrates with dot-net.
- pyQuil, QuTiP
  - Tool boxed for quantum computations implemented in Python.

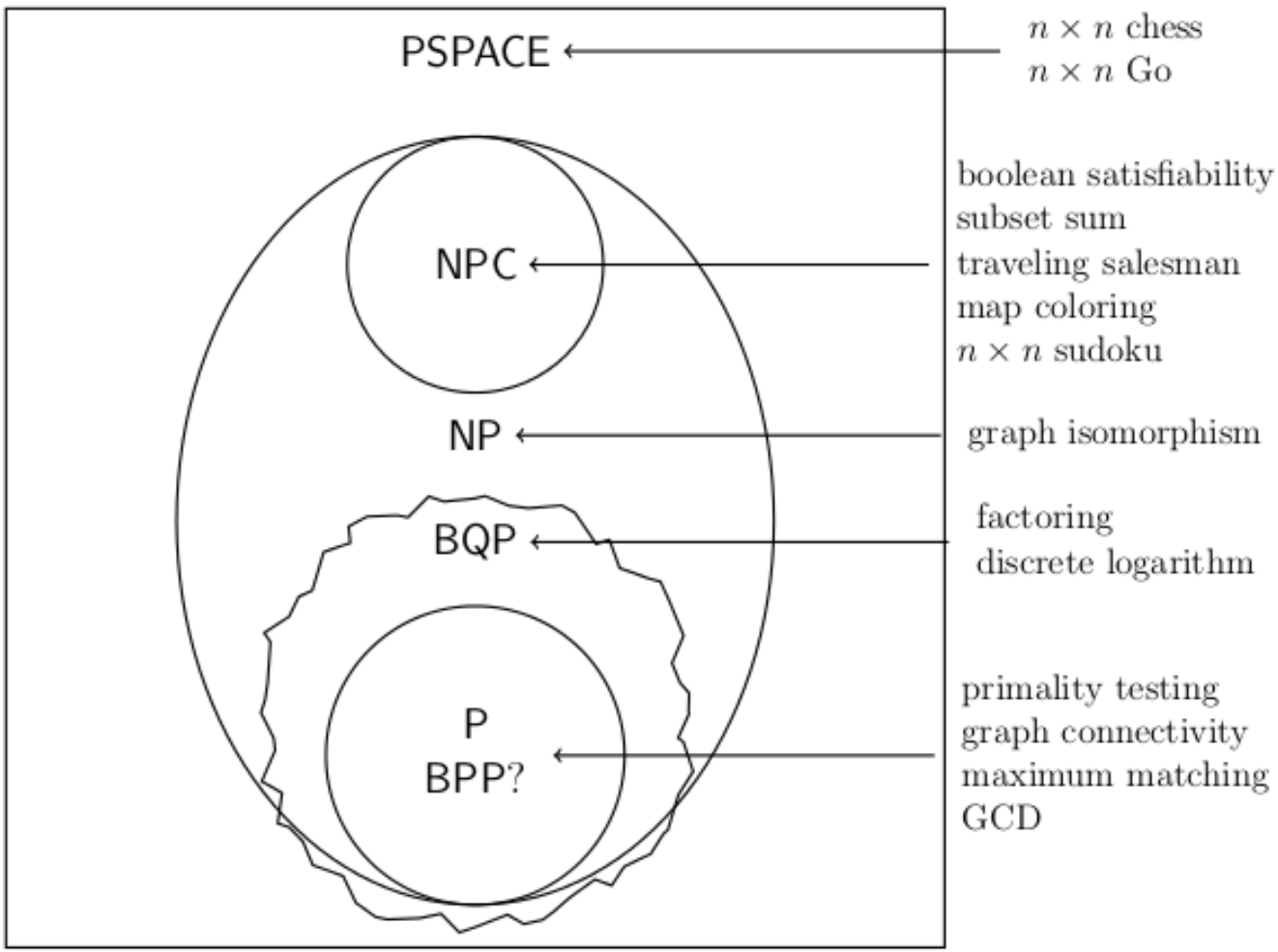
And others...

A touch of Quantum Algorithms

Performing Quantum Computations

Conjectured complexity classes

[Strubell, 2011]



A touch of Quantum Algorithms

Performing Quantum Computations

Grover's algorithm

Performs a search for an element over an n-element unordered list

- Best know algorithm runs $O(n)$
- Quantum algorithm runs $O(\sqrt{n})$

Algorithm outline:

1. Put all possible 2^n states in equal superposition. All possible element is equally like to be the one be search for.
2. Perform quantum operations such that measurement will give correct outcome with probability higher that $1/2$.

This is the general approach of quantum algorithms.

Conclusion

Recap

Quantum Circuit Model

Performing Quantum Computations

Conclusion

Conclusion

Conclusion

- Even with quantum computers we build abstractions to be able to work with the underlying implementation:
  - quantum and reversible circuits,
  - description languages,
  - algorithmic descriptions.
- Part of doing quantum computations is successfully working with these.
- Our interaction in the model is still very basic.
- There exist quantum algorithms that performs asymptotically better than known conventional algorithms.

Thank you

?

Bibliography I


Cuccaro, S. A., Draper, T. G., Kutin, S. A., and Moulton, D. P. (2005).
A new quantum ripple-carry addition circuit.
arXiv:quant-ph/0410184v1.


Feynman, R. P. (1985).
Quantum mechanical computers.
Optics News, 11:11–20.


Fredkin, E. and Toffoli, T. (1982).
Conservative logic.
International Journal of Theoretical Physics, 21(3-4):219–253.


Strubell, E. (2011).
An introduction to quantum algorithms.


Toffoli, T. (1980).
Reversible computing.
In de Bakker, J. W. and van Leeuwen, J., editors, *Automata, Languages and Programming. Proceedings*, volume 85 of *Lecture Notes in Computer Science*, pages 632–644. Springer-Verlag.


Vedral, V., Barenco, A., and Ekert, A. (1996).
Quantum networks for elementary arithmetic operations.
Physical Review A, 54(1):147–153.