igloo CONF.fi

SUMMER EDITION

Creating a quantum algorithm

using Microsoft Q# and Azure Quantum

Johnny Hooyberghs

Here’s Johnny

Johnny Hooyberghs

Passionate Developer

Principal Software Consultant/Architect (.NET)

Microsoft MVP, Developer Technologies

Operational Manager at Involved

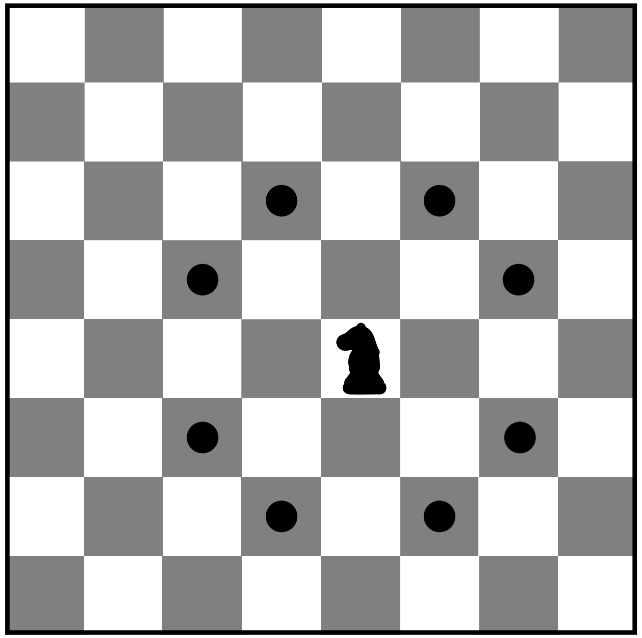
Johnny Hooyberghs

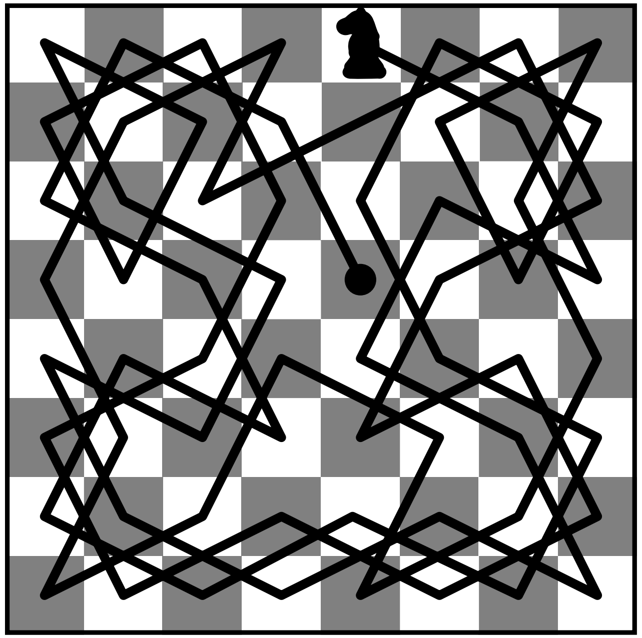
@djohnnieke

johnny.hooyberghs@involved.be

https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

Why Quantum Computing?

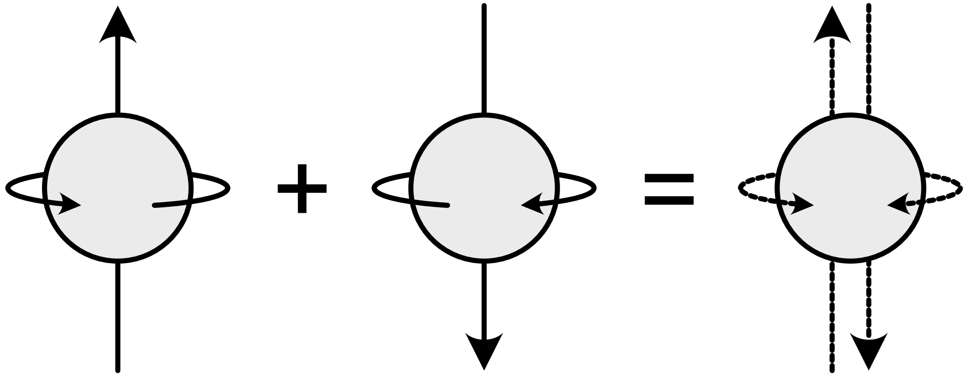


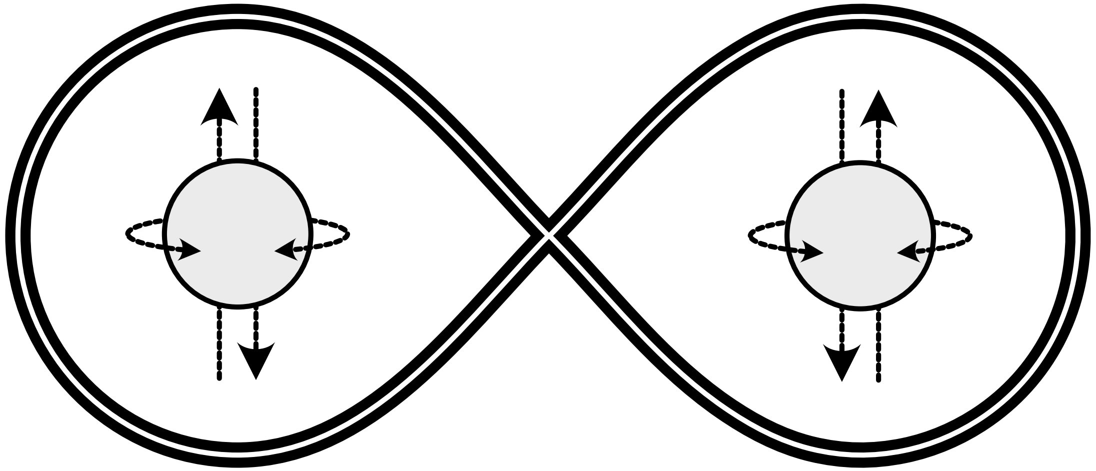


https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

Superposition and Entanglement

- Quantum mechanics describes superposition and entanglement of quantum particles
- Quantum computing can use these phenomena to its advantage





Why Quantum Computing?

- Security
- Communication
- Drug Development
- AI and/or Machine Learning
- Saving the polar bears!

https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

Bits vs. Qubits

0

1

Bits vs. Qubits

100110

Bits vs. Qubits

$$|0\rangle \qquad |1\rangle$$

Bits vs. Qubits

$$|100110\rangle$$

Quantum state

$$\boldsymbol{\alpha}|0\rangle + \boldsymbol{\beta}|1\rangle$$

Quantum state

$$\boldsymbol{\alpha}|0\rangle + \boldsymbol{\beta}|1\rangle$$

$$|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2 = 1$$

Quantum state

$$\boldsymbol{\alpha}|0\rangle + \boldsymbol{\beta}|1\rangle$$

$$|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2 = 1$$

$$\boldsymbol{\alpha} = a + bi$$

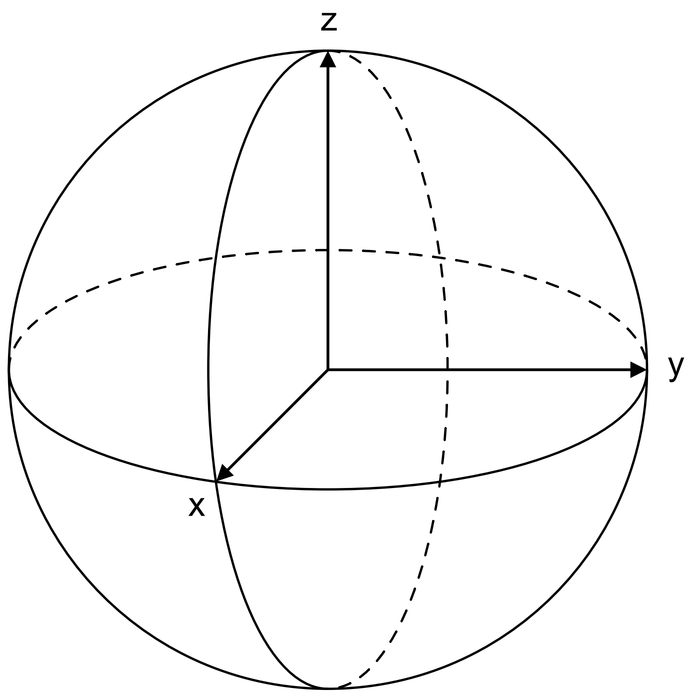
$$\boldsymbol{\beta} = c + di$$

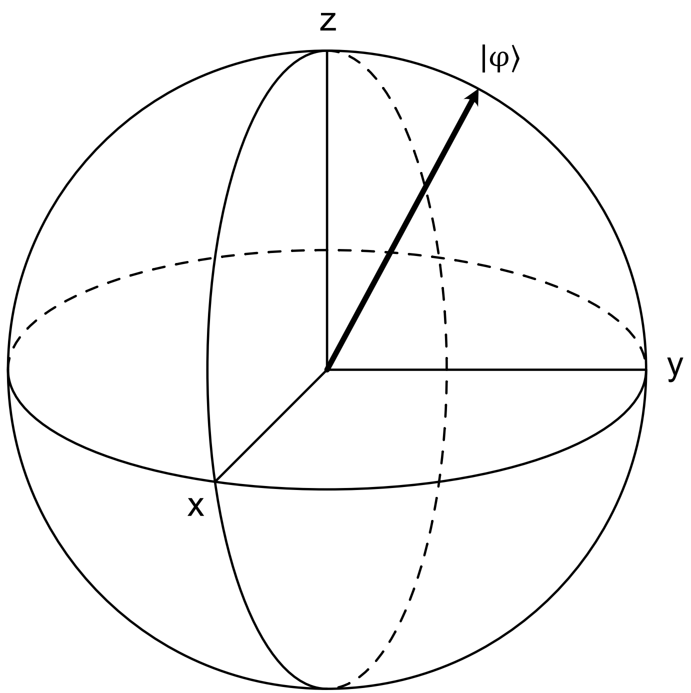
https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

Quantum state

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

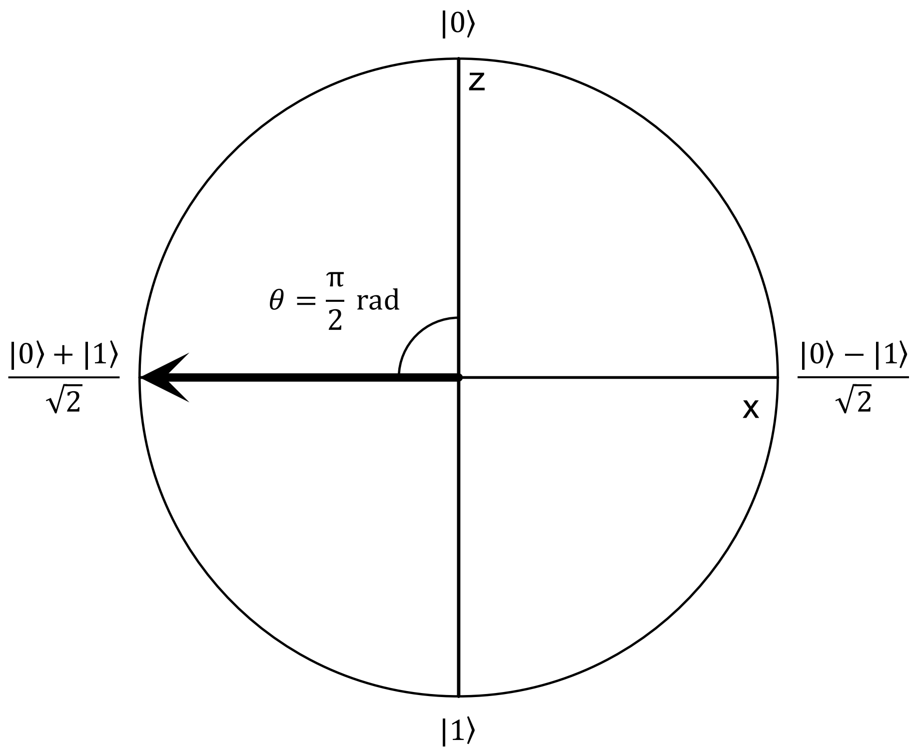
Quantum state

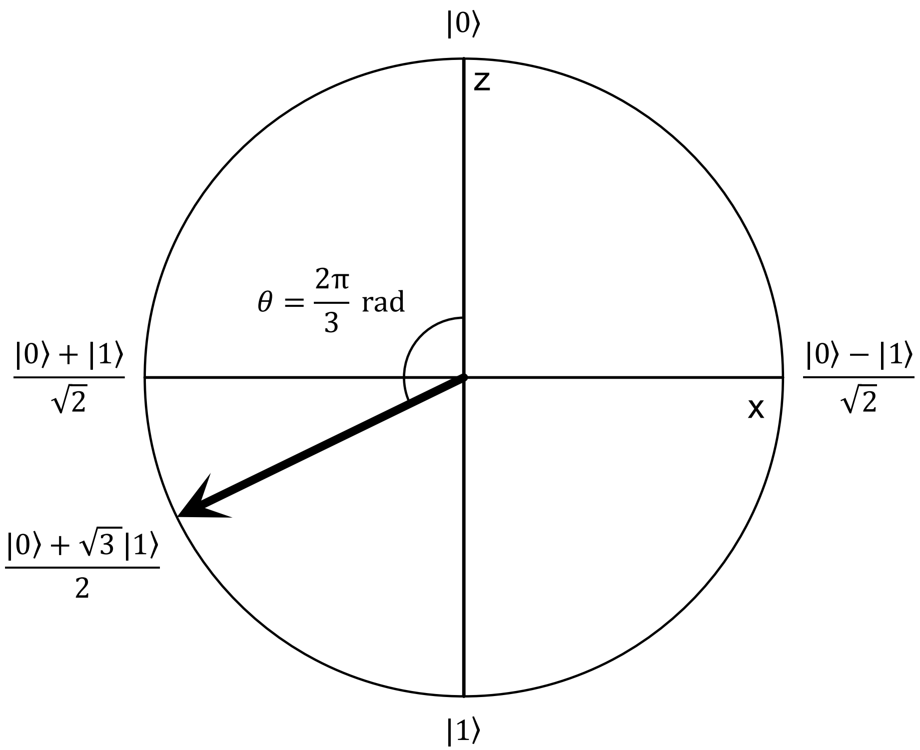




https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

Quantum state





https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

Quantum state

2 Qubit system (4 probabilities):

$$\boldsymbol{\alpha}|00\rangle + \boldsymbol{\beta}|01\rangle + \boldsymbol{\gamma}|10\rangle + \boldsymbol{\delta}|11\rangle$$

Quantum state

2 Qubit system (4 probabilities):

$$\boldsymbol{\alpha}|00\rangle + \boldsymbol{\beta}|01\rangle + \boldsymbol{\gamma}|10\rangle + \boldsymbol{\delta}|11\rangle$$

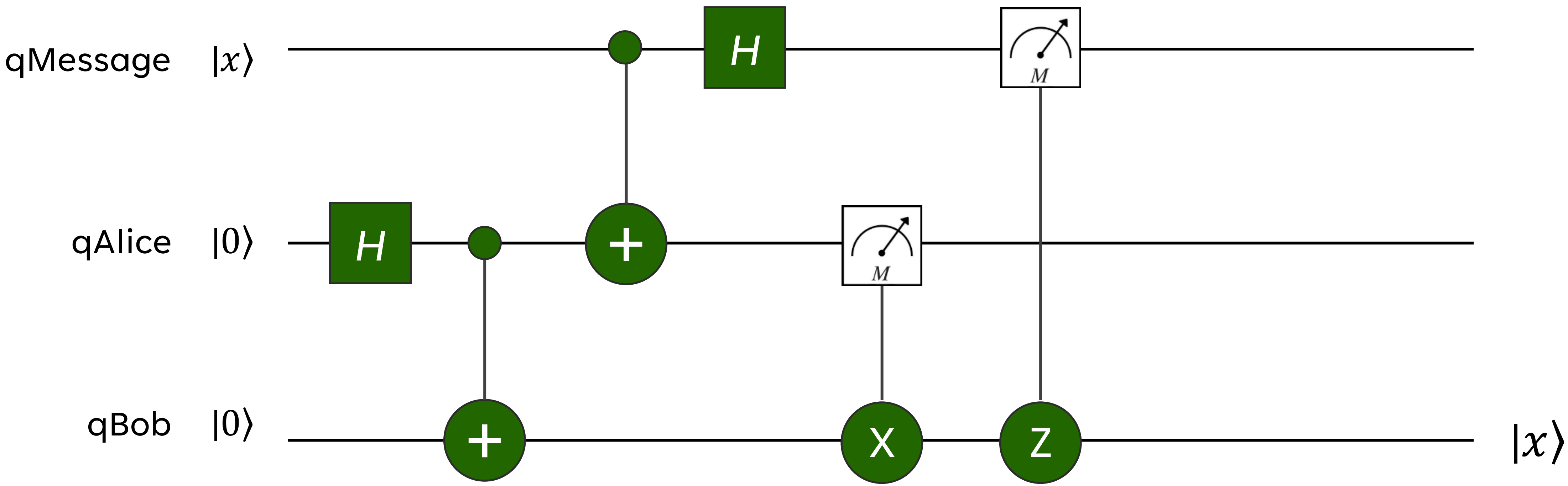
3 Qubit system (8 probabilities):

$$\boldsymbol{\alpha}|000\rangle + \boldsymbol{\beta}|001\rangle + \boldsymbol{\gamma}|010\rangle + \boldsymbol{\delta}|011\rangle + \boldsymbol{\varepsilon}|100\rangle + \boldsymbol{\epsilon}|110\rangle + \boldsymbol{\zeta}|101\rangle + \boldsymbol{\eta}|111\rangle$$

4 Qubit system (16 probabilities):

$$\boldsymbol{\alpha}|0000\rangle + \boldsymbol{\beta}|0001\rangle + \boldsymbol{\gamma}|0010\rangle + \boldsymbol{\delta}|0011\rangle + \boldsymbol{\varepsilon}|0100\rangle + \boldsymbol{\epsilon}|0110\rangle + \boldsymbol{\zeta}|0101\rangle + \boldsymbol{\eta}|0111\rangle + \boldsymbol{\theta}|1000\rangle$$
$$+ \boldsymbol{\vartheta}|1001\rangle + \boldsymbol{\iota}|1010\rangle + \boldsymbol{\kappa}|1011\rangle + \boldsymbol{\lambda}|1100\rangle + \boldsymbol{\mu}|1110\rangle + \boldsymbol{\nu}|1101\rangle + \boldsymbol{\xi}|1111\rangle$$

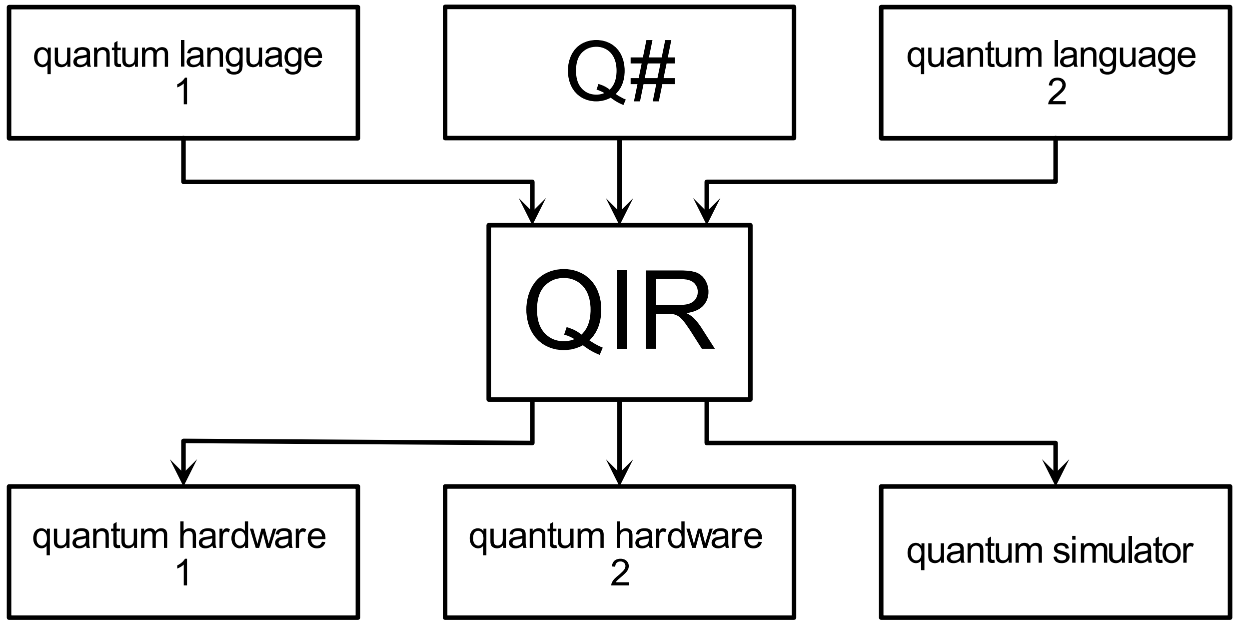
Teleportation



https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

Azure Quantum

- Quantum in the cloud
  - Optimization
  - Machine Learning
  - Quantum Simulation
- Access to quantum hardware
  - Microsoft (Topological)
  - IonQ & Quantinuum (Ion Traps)
  - QCI & Rigetti (Superconducting)
  - Pasqal (Neutral Atom)
- Q# & QDK
  - Quantum Intermediate Representation (QIR)



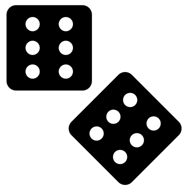
https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

CHSH game

Random bit **X**

Jääkarhu

Output bit **A**



Random bit **Y**

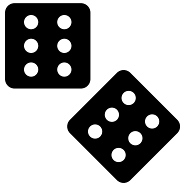
Pingviini

Output bit **B**

$$X \cdot Y = A \oplus B$$

https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

CHSH game



Random bit **X**

Jääkarhu

Output bit **A**

Random bit **Y**

Pingviini

Output bit **B**

X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

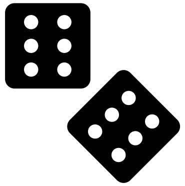
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

$$X \cdot Y = A \oplus B$$

https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

CHSH game

Random bit **X**



Random bit **Y**

X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

Jääkarhu

Output bit **A**

Pingviini

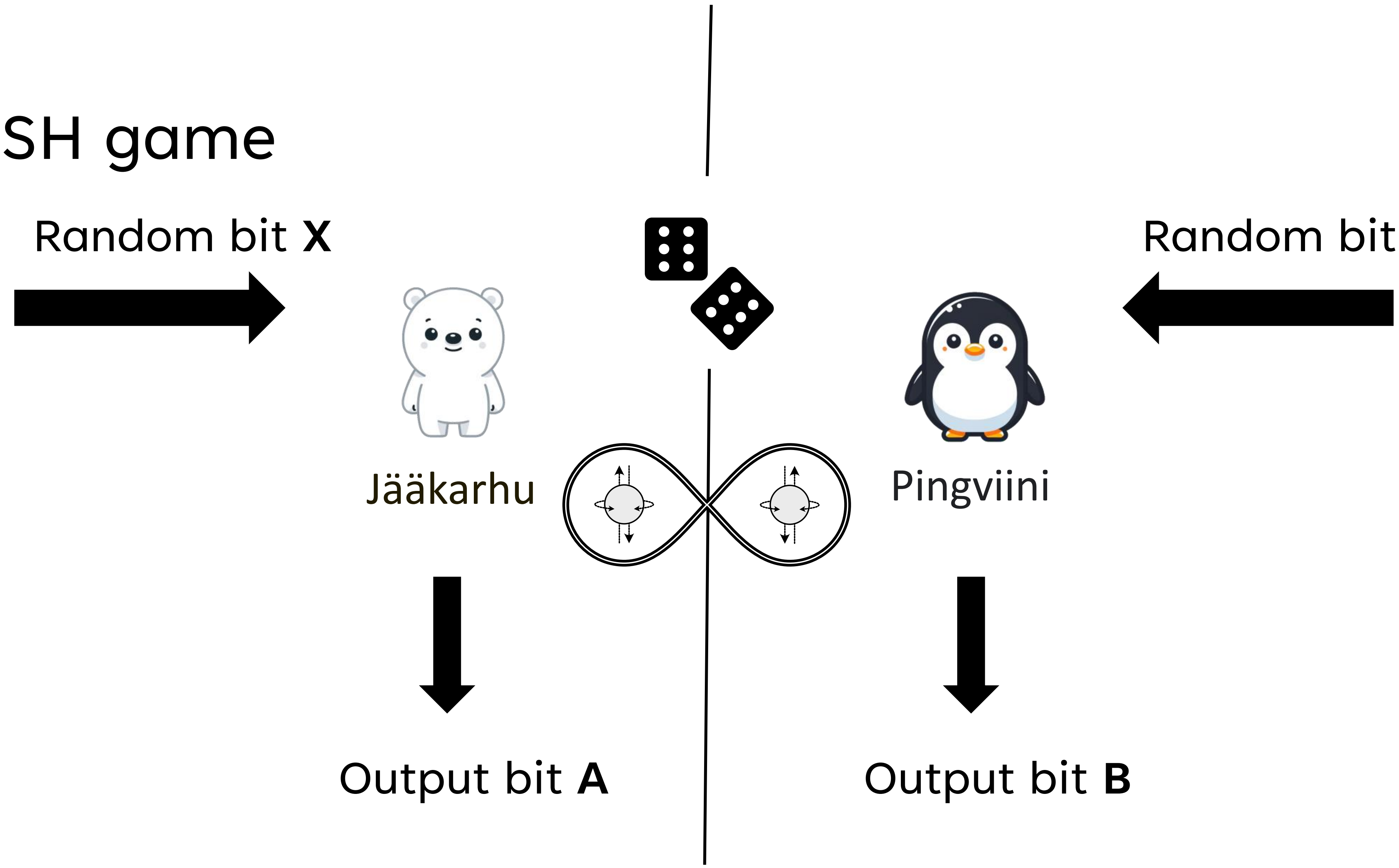
Output bit **B**

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

$$X \cdot Y = A \oplus B$$

https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

CHSH game



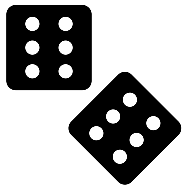
$$X \cdot Y = A \oplus B$$

https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

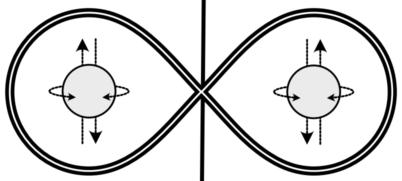
CHSH game

Random bit **X**

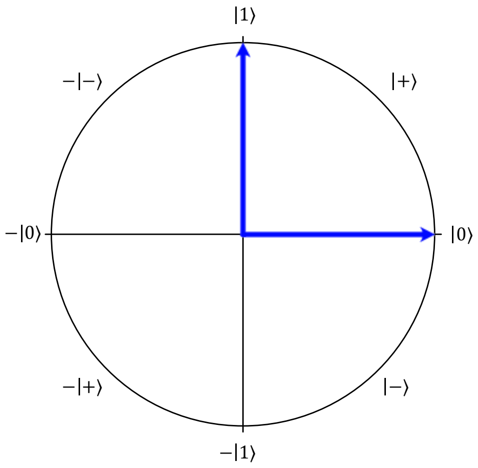
Random bit **Y**

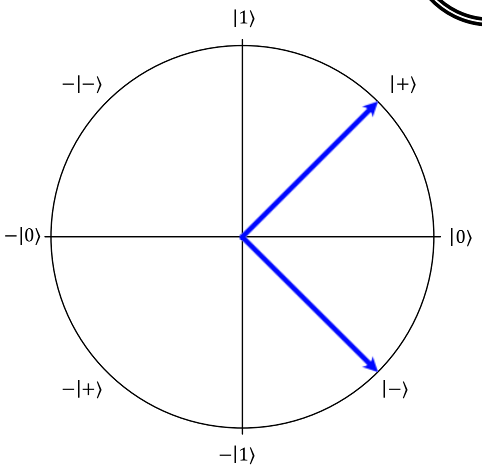


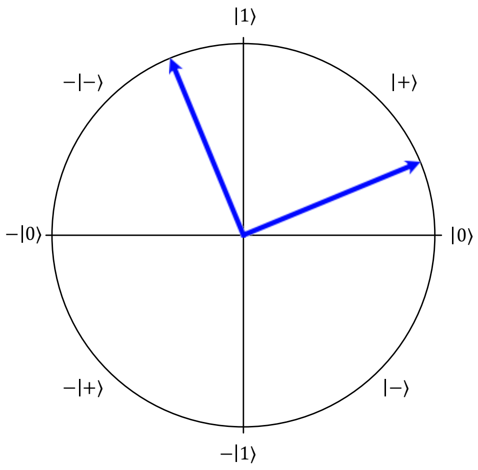
Jääkarhu

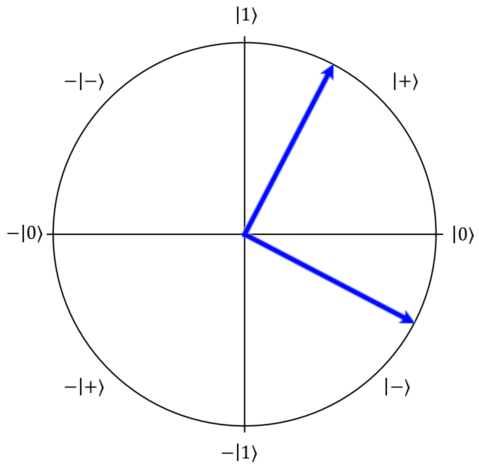


Pingviini



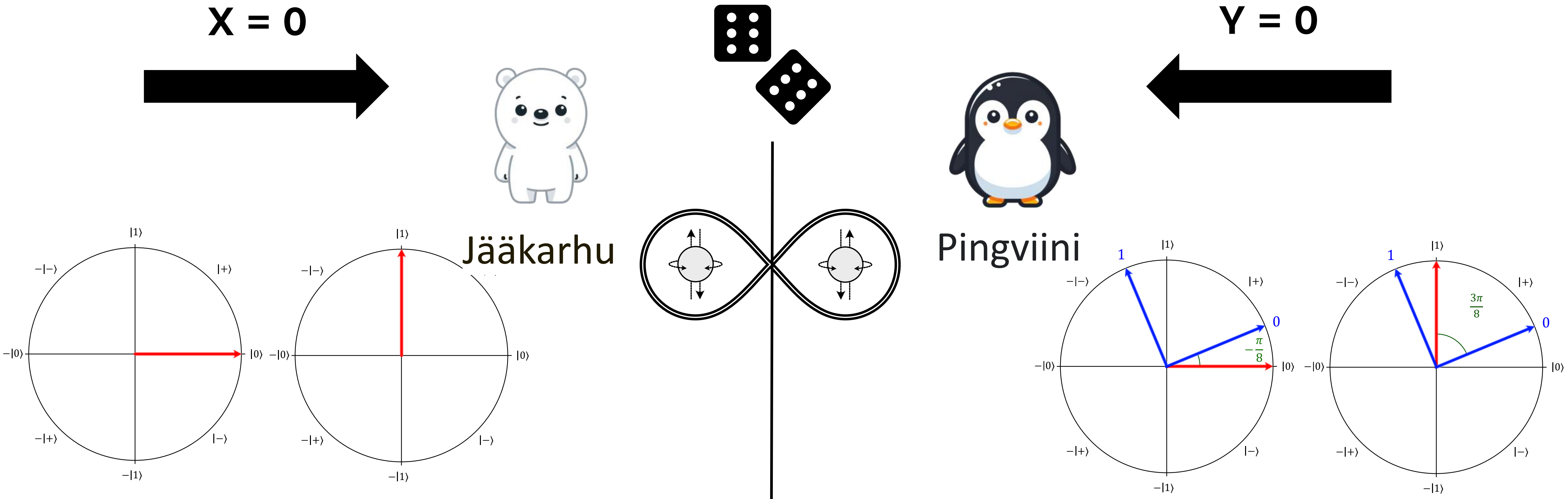






https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024

CHSH game

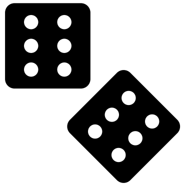


Jääkarhu outputs	Pingviini’s qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$\lvert 0\rangle$	$\cos^2\left(-\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(-\frac{\pi}{8}\right) \approx 0.15$
1	$\lvert 1\rangle$	$\cos^2\left(\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{3\pi}{8}\right) \approx 0.85$

CHSH game

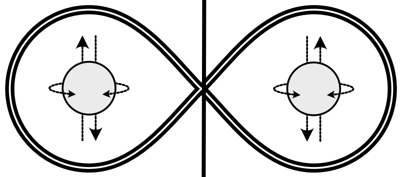
X = 0

Y = 0



Jääkarhu

Pingviini

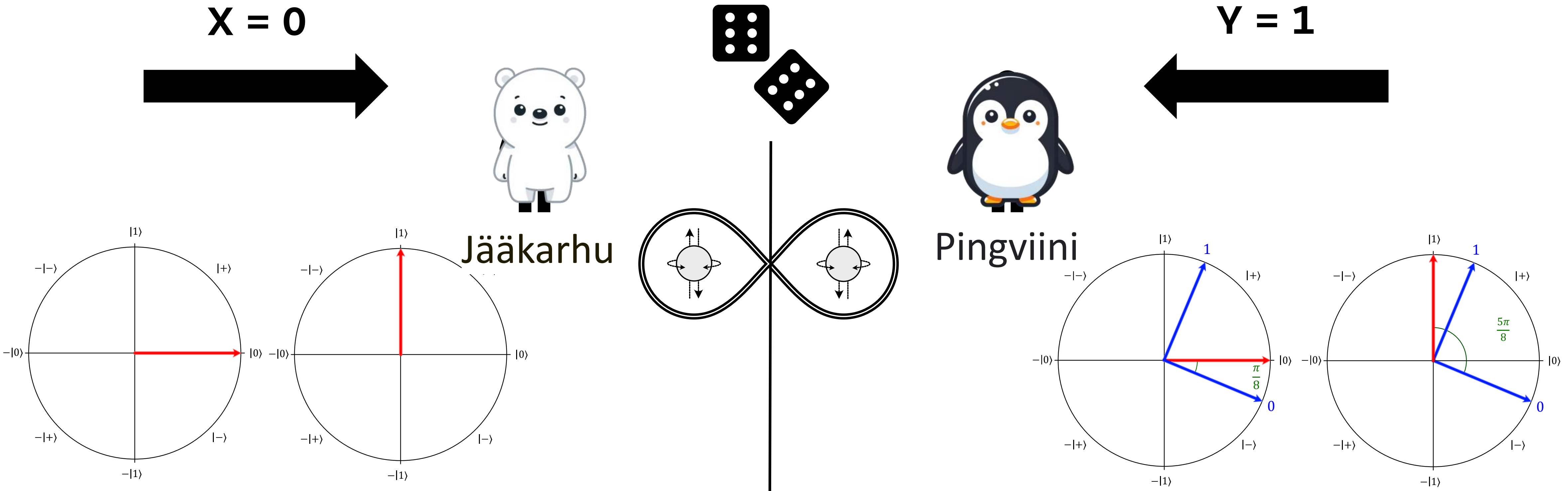


X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$\|0\rangle$	$\cos^2\left(-\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(-\frac{\pi}{8}\right) \approx 0.15$
1	$\|1\rangle$	$\cos^2\left(\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{3\pi}{8}\right) \approx 0.85$

CHSH game



Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$\vert 0\rangle$	$\cos^2\left(\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(\frac{\pi}{8}\right) \approx 0.15$
1	$\vert 1\rangle$	$\cos^2\left(\frac{5\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{5\pi}{8}\right) \approx 0.85$

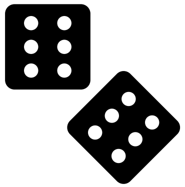
CHSH game

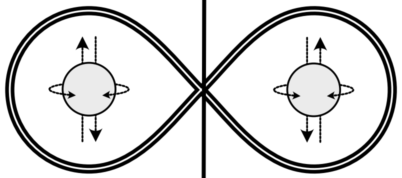
X = 0

Y = 1

X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

Jääkarhu



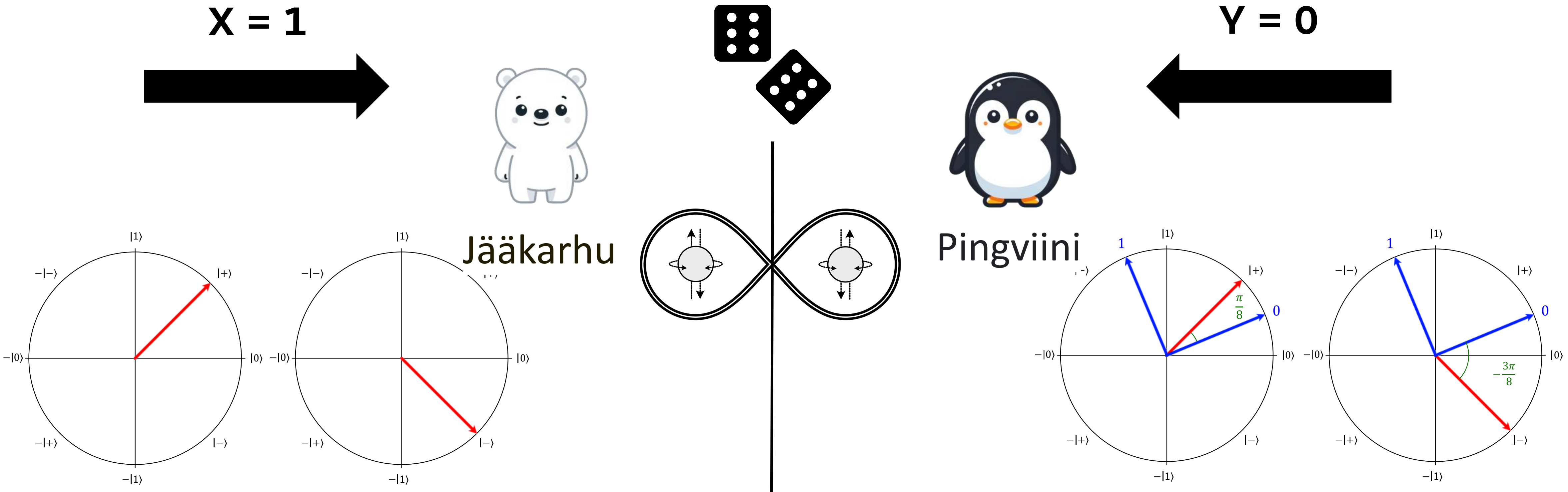


Pingviini

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$\vert 0\rangle$	$\cos^2\left(\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(\frac{\pi}{8}\right) \approx 0.15$
1	$\vert 1\rangle$	$\cos^2\left(\frac{5\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{5\pi}{8}\right) \approx 0.85$

CHSH game

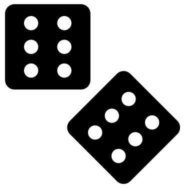


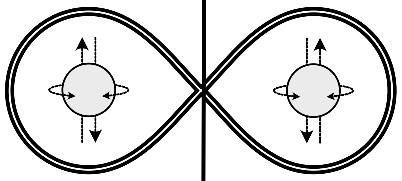
Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$\vert+\rangle$	$\cos^2\left(\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(\frac{\pi}{8}\right) \approx 0.15$
1	$\vert-\rangle$	$\cos^2\left(-\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(-\frac{3\pi}{8}\right) \approx 0.85$

CHSH game

X = 1

Jääkarhu





Pingviini

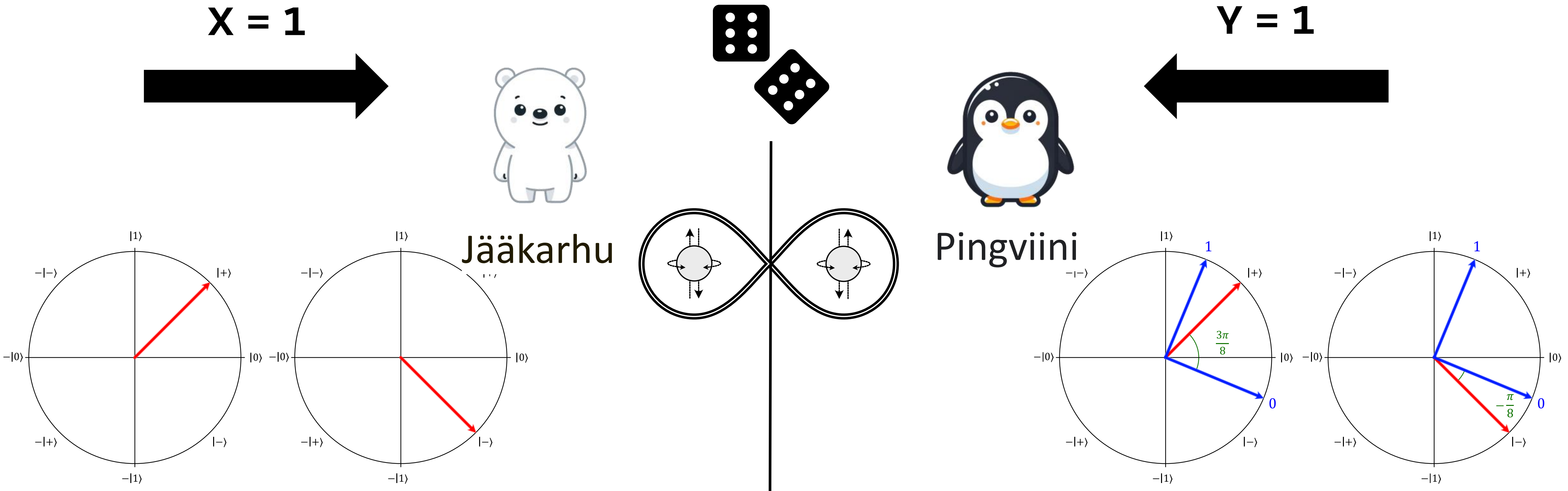
Y = 0

X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Jääkarhu outputs	**Pingviini's qbit**	**Pingviini outputs 0 with probability**	**Pingviini outputs 1 with probability**
0	$\vert + \rangle$	$\cos^2\left(\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(\frac{\pi}{8}\right) \approx 0.15$
1	$\vert - \rangle$	$\cos^2\left(-\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(-\frac{3\pi}{8}\right) \approx 0.85$

CHSH game



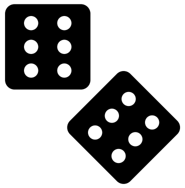
Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$\lvert+\rangle$	$\cos^2\left(\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{3\pi}{8}\right) \approx 0.85$
1	$\lvert-\rangle$	$\cos^2\left(-\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(-\frac{\pi}{8}\right) \approx 0.15$

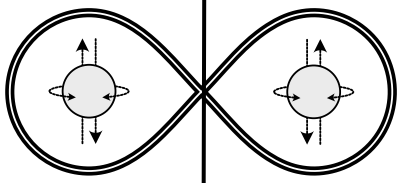
CHSH game

X = 1

Y = 1

Jääkarhu





Pingviini

X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$\vert+\rangle$	$\cos^2\left(\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{3\pi}{8}\right) \approx 0.85$
1	$\vert-\rangle$	$\cos^2\left(-\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(-\frac{\pi}{8}\right) \approx 0.15$

Microsoft Quantum Computing for Developers

…ction to quantum computing with the Microsoft Quantum Development Kit and Q# for

…mputing, but what does it mean to you as a software developer? With many new …d investment by some of the largest tech companies in the world to be the first to …rdware and platforms, it is no longer a tool in the distant future. Developers are at … that take advantage of QP through simulations. While the skill is of interest, for … implications still remains a mystery.

…d exploring important quantum concepts and apply them in practice through … Microsoft Quantum Development Kit. Theoretical knowledge about quantum …t, will be used to explain quantum computing topics, including quantum gates, …ally, take a tour of the new Azure Quantum.

…ge, to target quantum hardware. You will select your supporting language of … your quantum applications. Combined with just enough theoretical preparation, …dy to simulate basic quantum programs using Microsoft Visual Studio or Visual

…ndependent quantum tool set using the Microsoft Quantum Development Kit …de or Microsoft Visual Studio
…n mechanics required to start working on quantum computing
…ical concepts such as complex numbers, trigonometry, and linear algebra
…oft Quantum Development Kit on a Windows or Linux PC with Visual Studio Code or Microsoft …Studio
- Write quantum algorithms with the Microsoft Quantum Development Kit and Q#, supported by C# or Python
- Discover insights on important existing quantum algorithms such as Deutch, Deutch-Jozsa, and the fun CHSH-game
- Get introduced to quantum as a service using the Microsoft Azure Quantum preview cloud offering

This book is for developers who are interested in quantum computing, specifically those software developers who are planning on using quantum computers in the future. Basic imperative programming knowledge is useful to understand the syntax and structure found in the Q# programming language. Knowledge of Microsoft C# or Python is not required since these languages are only used to support the simulation of Q# on a classical computer.

Johnny Hooybergs is a consultant for Involved, a Belgium based company centered on the design, development, and delivery of custom made software, where his expertise has been on .NET architecture and backend development. Since 2020, Johnny is a Microsoft Most Valuable Professional (MVP) in the category of Developer Technologies. He has been passionate about .NET from its first release and possesses a deep knowledge of C#, .NET, .NET Core, ASP.NET, Entity Framework, Azure and ALM using the Microsoft Stack. He enjoys the occasional web development using JavaScript. For more than a decade, he has allocated a portion of his free time to teaching .NET and C# for the adult education institute CVO Antwerpen. When he is not working or teaching, he can be found gaming, scuba diving, learning to play the piano, traveling the world and visiting as many theme parks as possible.

ISBN 978-1-4842-7245-9
54999
9 781484 272459

Shelve in:
Microsoft

User level:
Beginning–Intermediate

Microsoft® Most Valuable Professional

Apress®
www.apress.com

Hooybergs Introducing Microsoft Quantum Computing for Developers Apress®

Introducing Microsoft Quantum Computing for Developers

Using the Quantum Development Kit and Q#

Johnny Hooybergs

Apress®

johnny.hooyberghs@involved.be

https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024