



Creating a quantum algorithm using Microsoft Q# and Azure Quantum




Johnny Hooyberghs

Here's Johnny



Johnny Hooyberghs

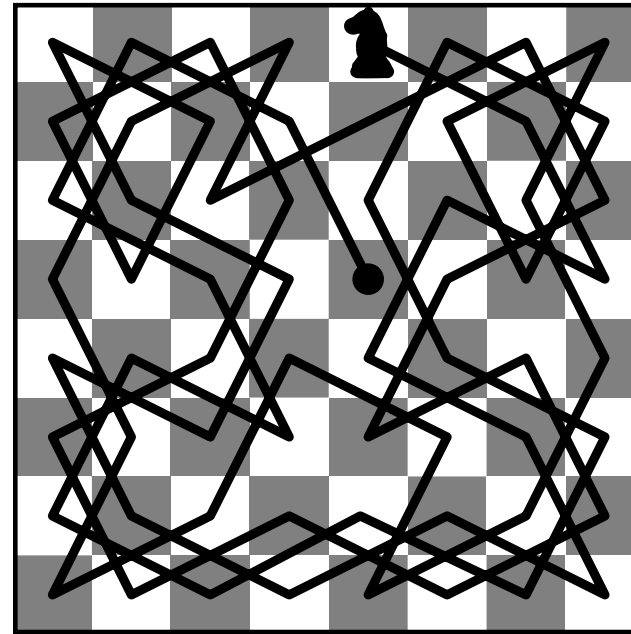
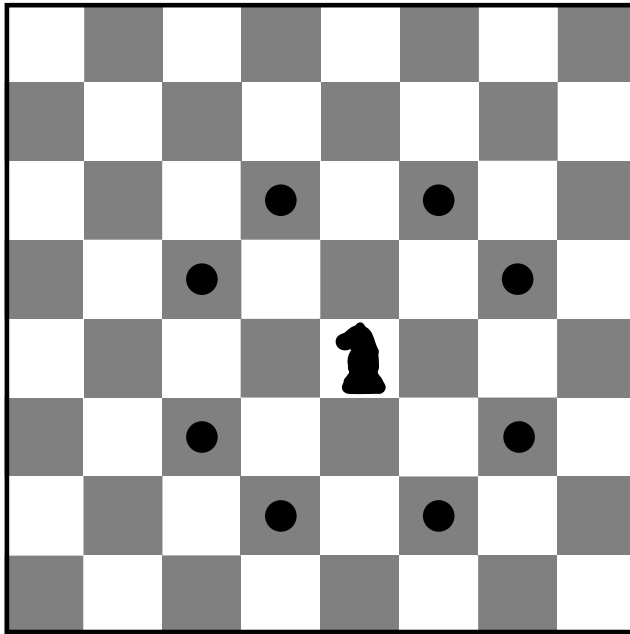
Passionate Developer
Principal Software Consultant/Architect (.NET)
Microsoft MVP, Developer Technologies
Operational Manager at Involved

 Johnny Hooyberghs
 @djohnnieke
 johnny.hooyberghs@involved.be



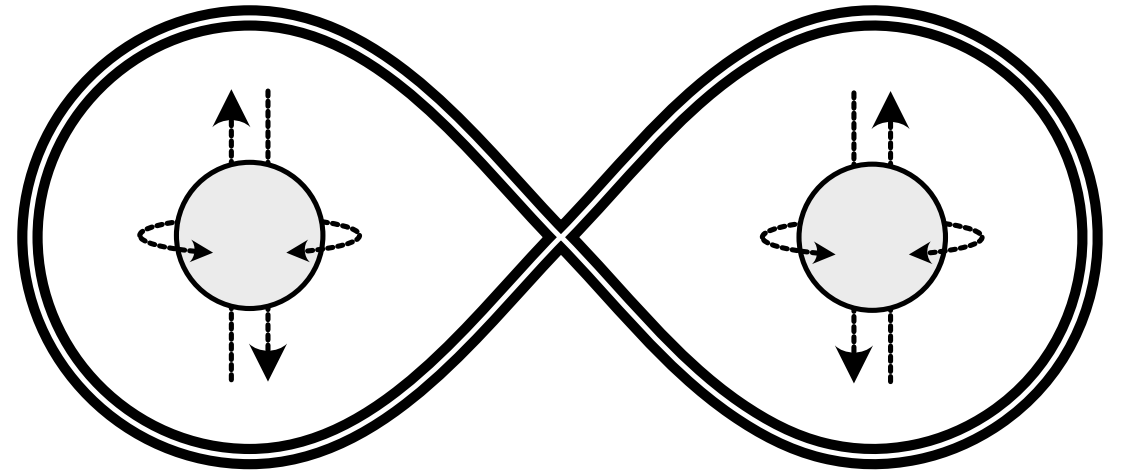
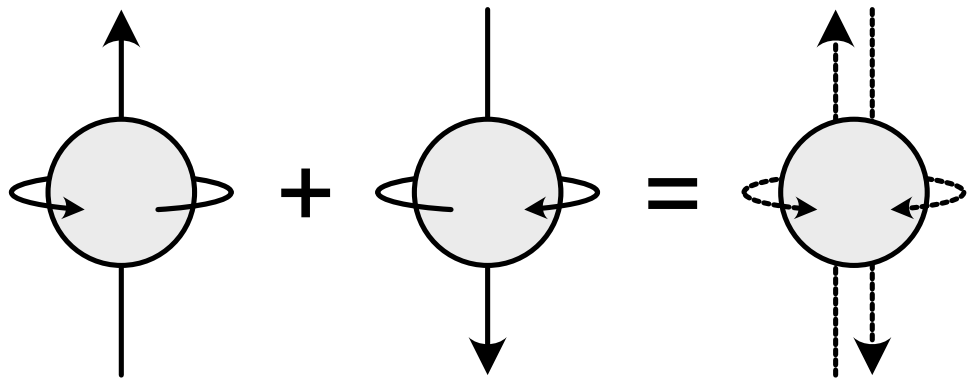
<https://github.com/Djohnnie/CreatingQuantumAlgorithmUsingQSharp-IglooConf-2024>

Why Quantum Computing?



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!



Bits vs. Qubits

0

1

Bits vs. Qubits

100110

Bits vs. Qubits

$|0\rangle$

$|1\rangle$

Bits vs. Qubits

|100110>

Quantum state

$$\alpha|0\rangle + \beta|1\rangle$$

Quantum state

$$\alpha|0\rangle + \beta|1\rangle$$
$$|\alpha|^2 + |\beta|^2 = 1$$

Quantum state

$$\alpha|0\rangle + \beta|1\rangle$$

$$|\alpha|^2 + |\beta|^2 = 1$$

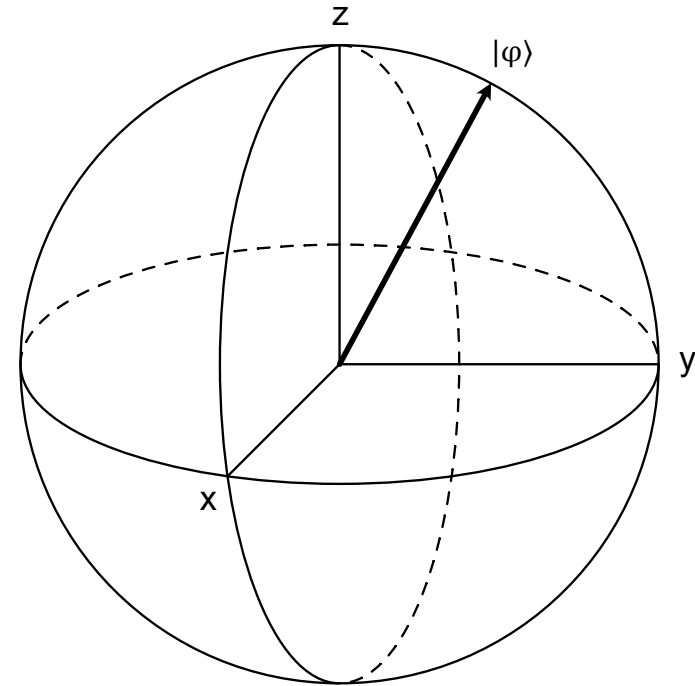
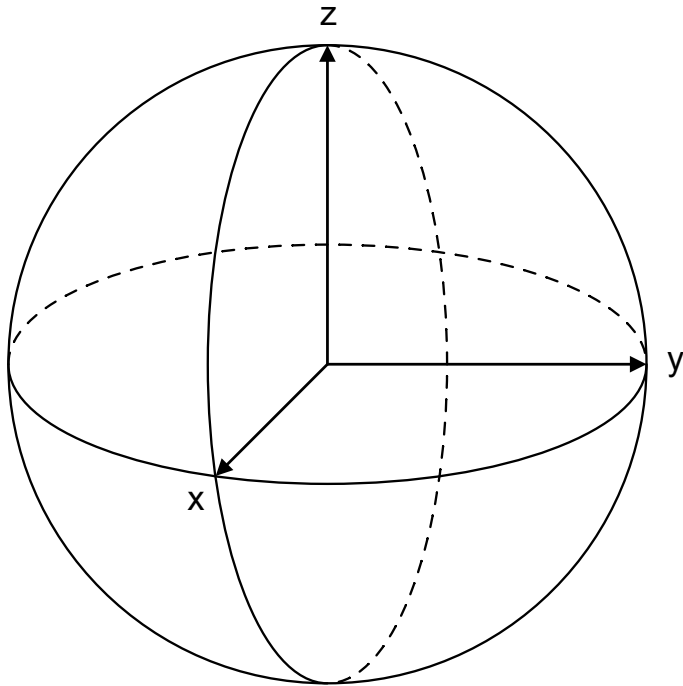
$$\alpha = a + bi$$

$$\beta = c + di$$

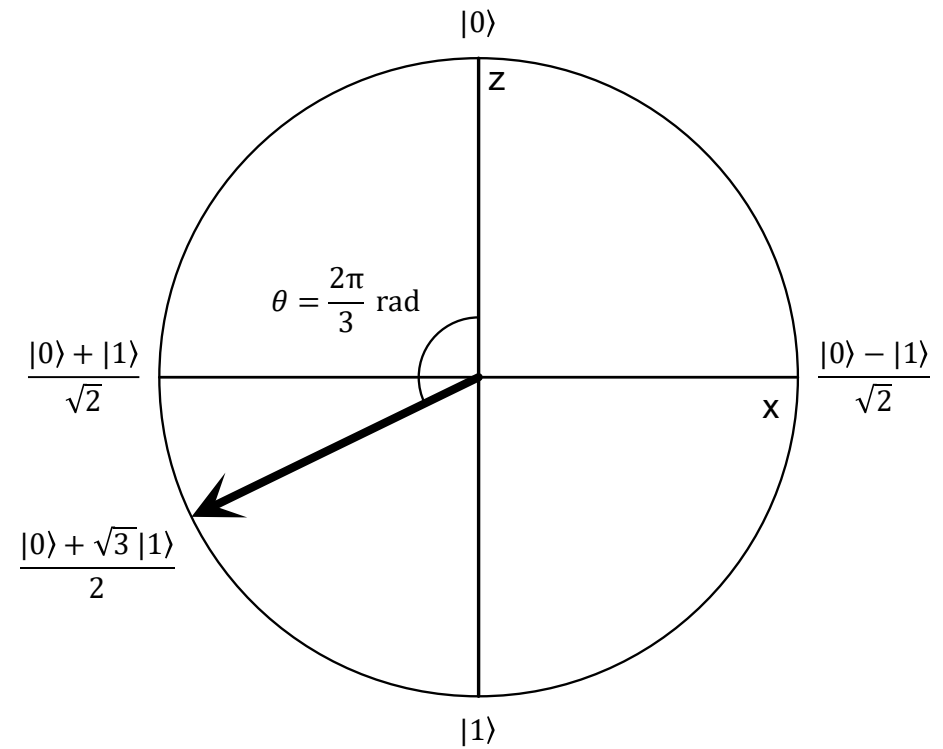
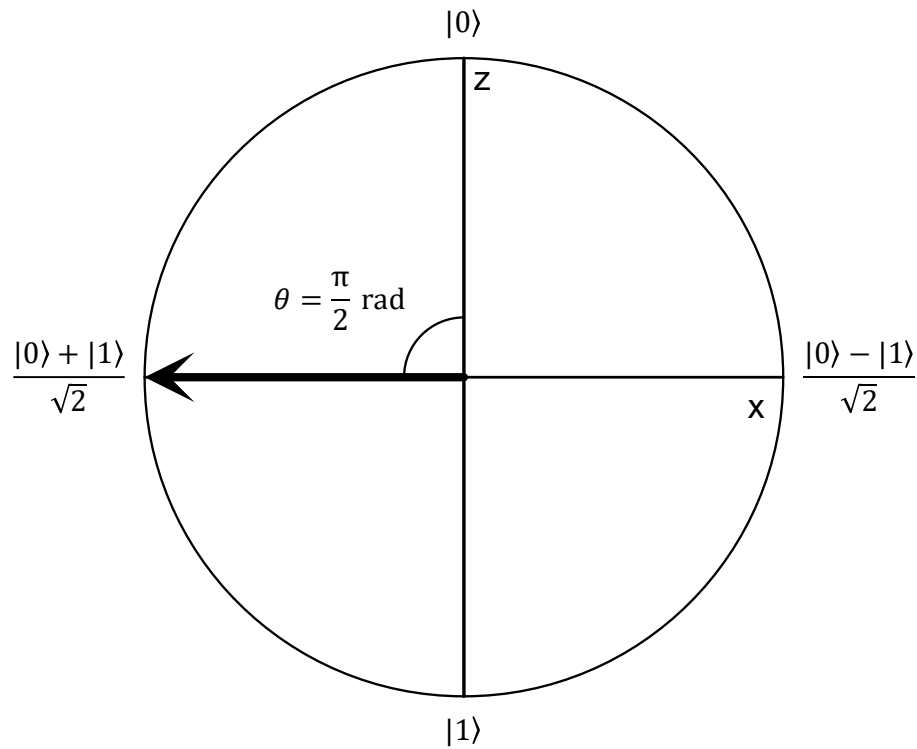
Quantum state

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

Quantum state



Quantum state



Quantum state

2 Qubit system (4 probabilities):

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

Quantum state

2 Qubit system (4 probabilities):

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

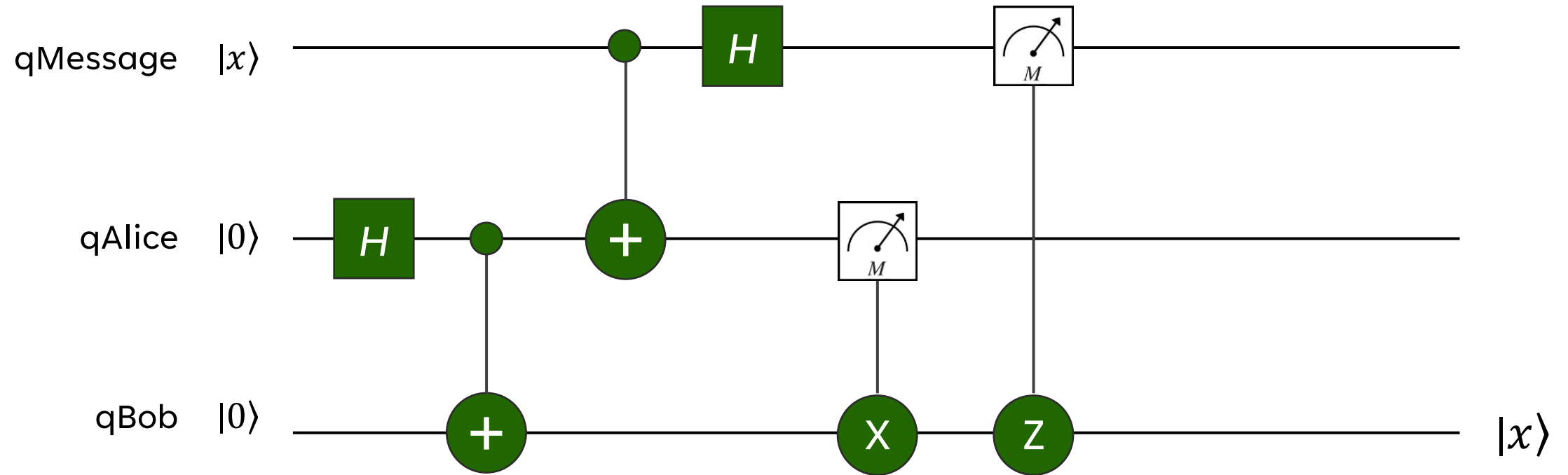
3 Qubit system (8 probabilities):

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

4 Qubit system (16 probabilities):

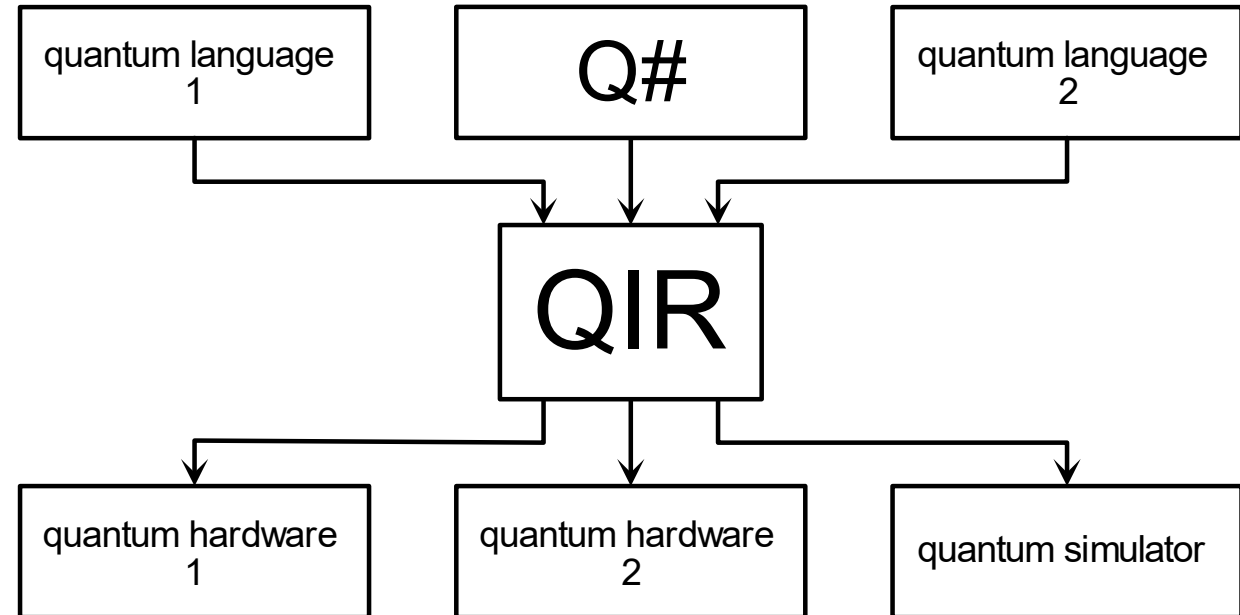
$$\begin{aligned} &\alpha|0000\rangle + \beta|0001\rangle + \gamma|0010\rangle + \delta|0011\rangle + \epsilon|0100\rangle + \zeta|0101\rangle + \eta|0110\rangle + \theta|0111\rangle + \iota|1000\rangle \\ &+ \kappa|1001\rangle + \lambda|1010\rangle + \mu|1011\rangle + \nu|1100\rangle + \xi|1101\rangle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

Teleportation

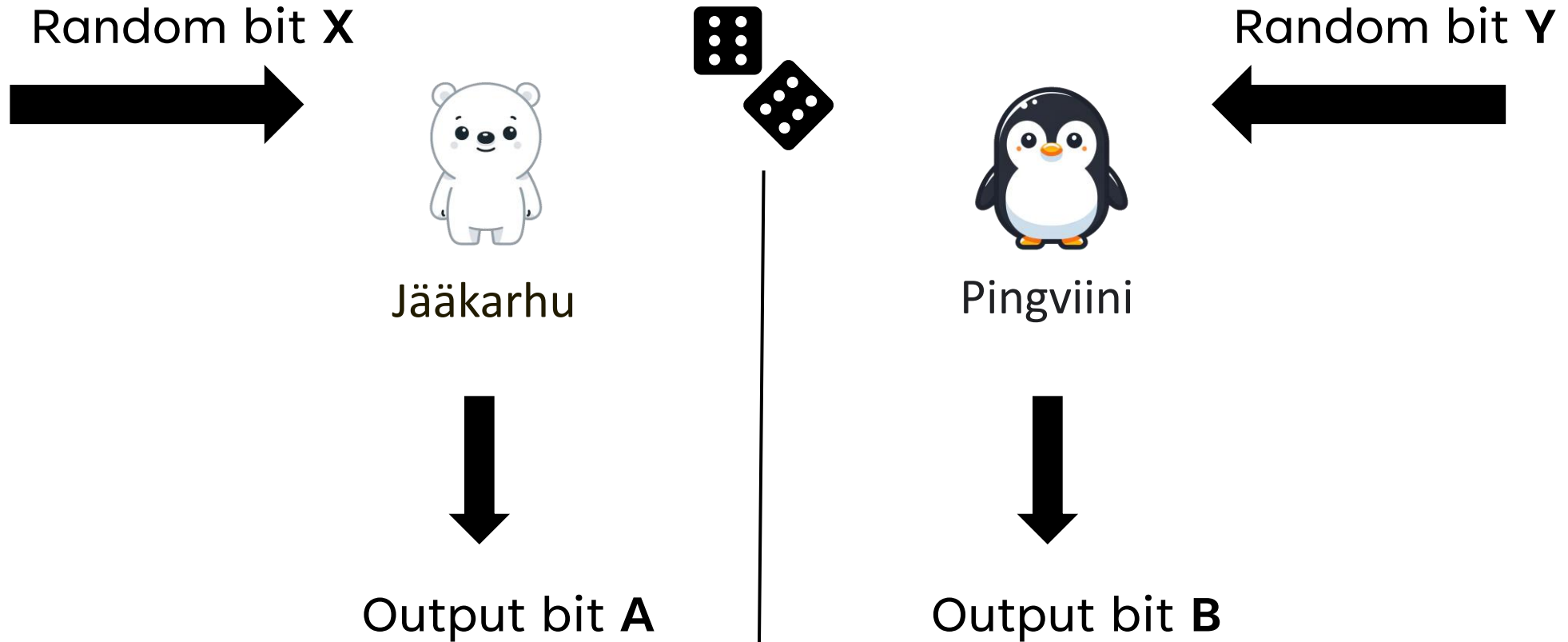


Azure Quantum

- 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)



CHSH game



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

CHSH game

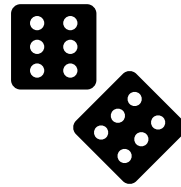
Random bit X



Jääkarhu



Output bit A



Random bit Y



Pingviini



Output bit B

X	Y	$X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

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

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

CHSH game

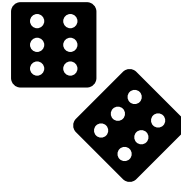
Random bit X



Jääkarhu



Output bit A



Pingviini



Output bit B

Random bit Y

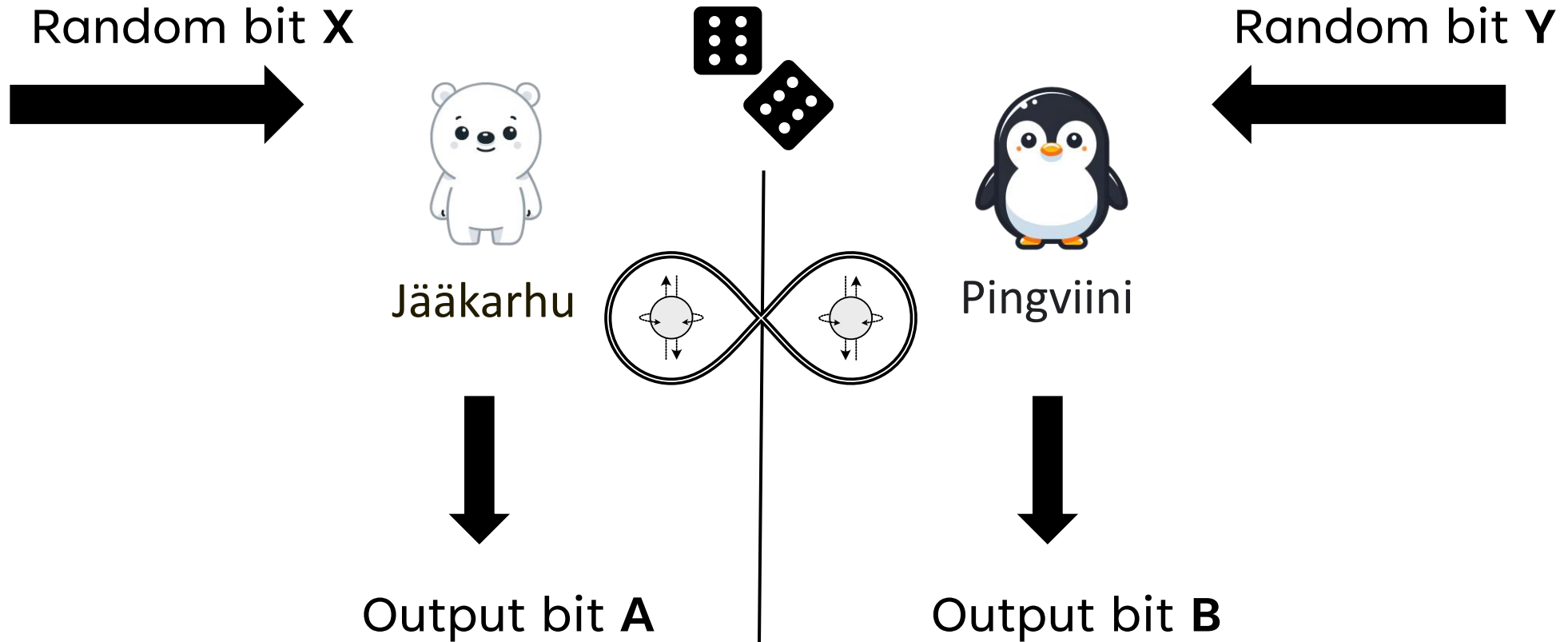


X	Y	$X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

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

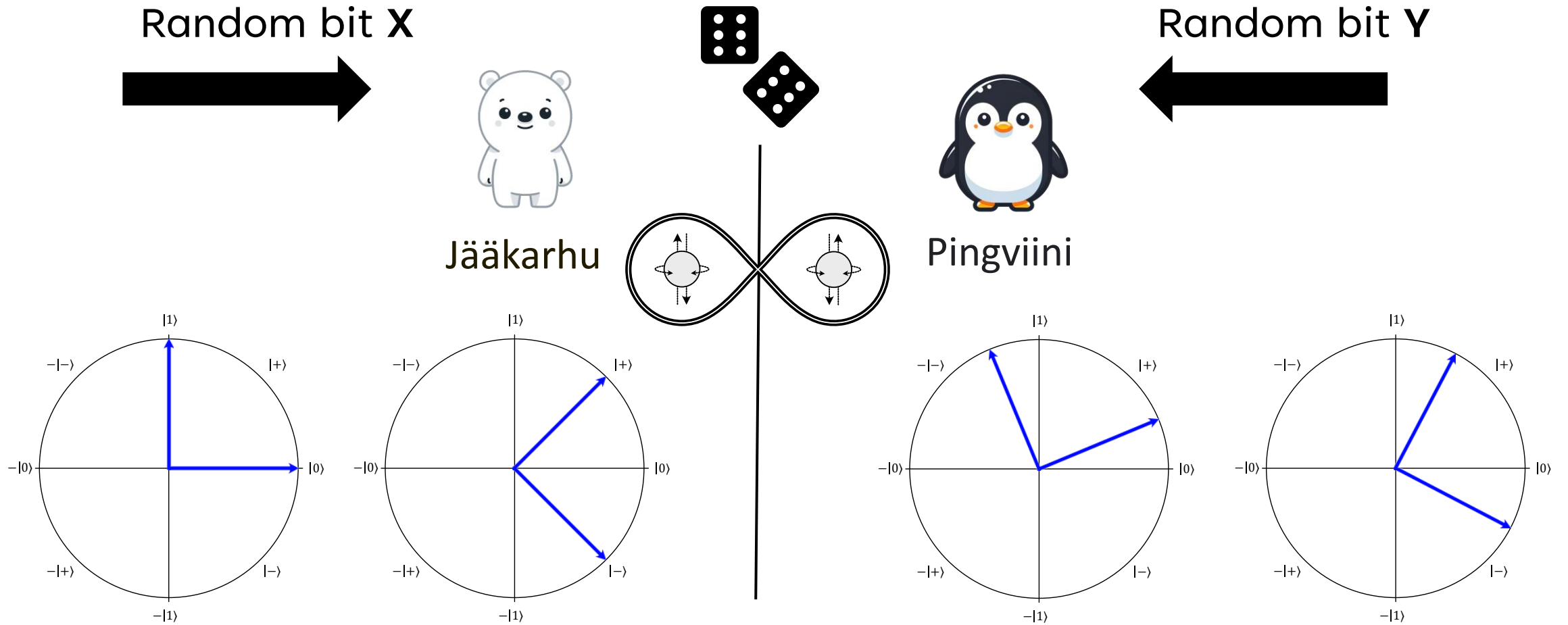
$$X \cdot Y = A \oplus B$$

CHSH game




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

CHSH game

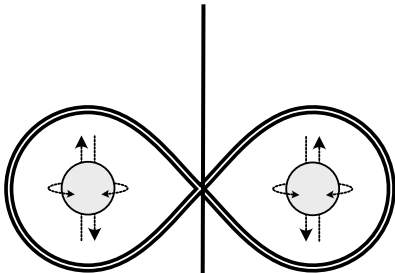
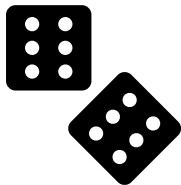
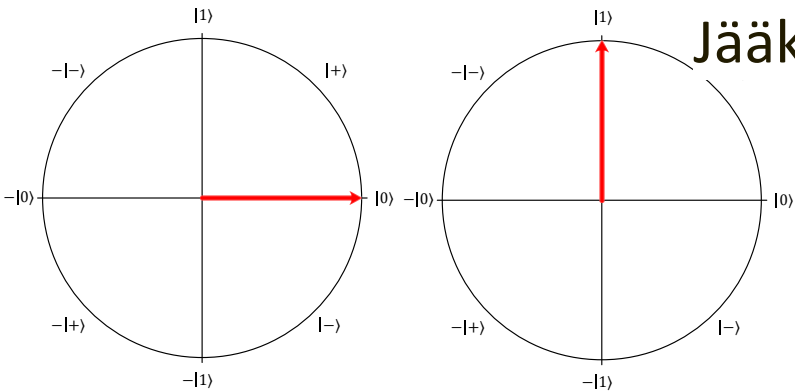


CHSH game

X = 0

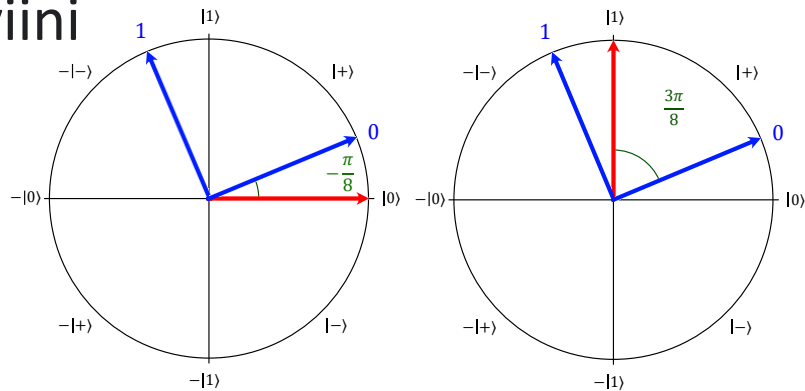



Jääkarhu



Pingviini

Y = 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

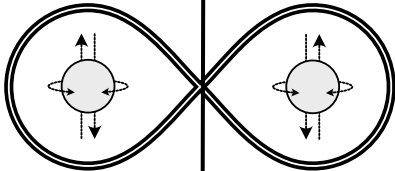
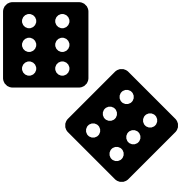
$X = 0$



X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1



Jääkarhu



Pingviini

$Y = 0$




A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

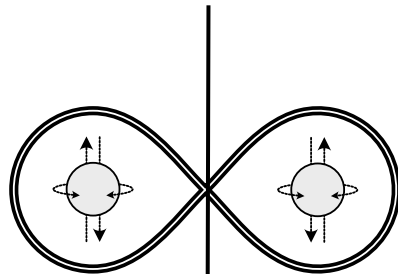
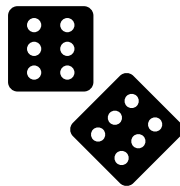
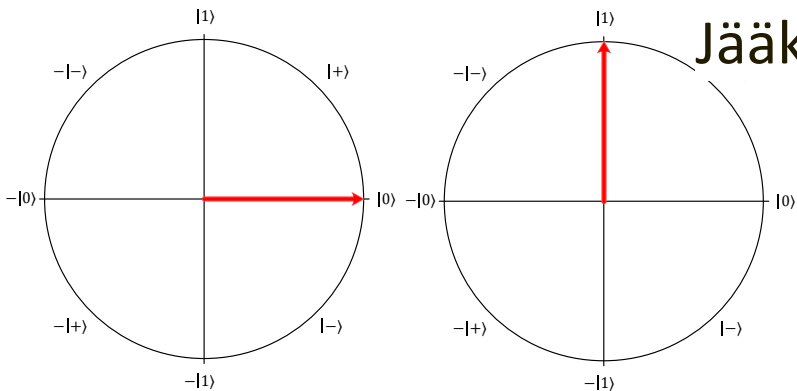
Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$ 0\rangle$	$\cos^2\left(-\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(-\frac{\pi}{8}\right) \approx 0.15$
1	$ 1\rangle$	$\cos^2\left(\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{3\pi}{8}\right)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

CHSH game

X = 0

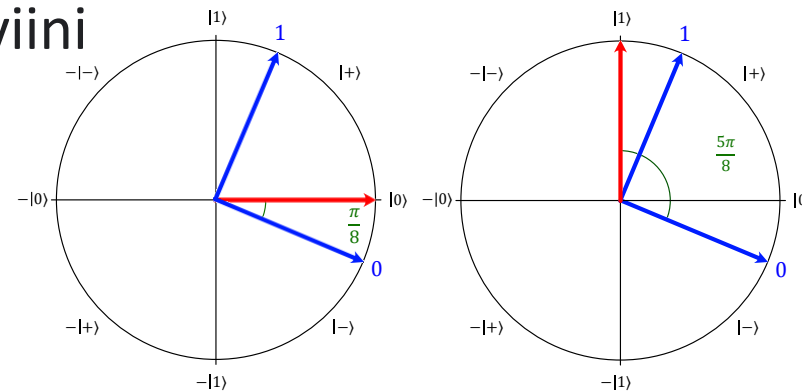



Jääkarhu



Pingviini

Y = 1



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$	$\cos^2\left(\frac{5\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{5\pi}{8}\right) \approx 0.85$

CHSH game

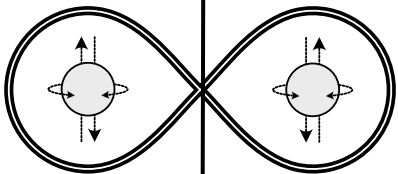
X = 0



X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1



Jääkarhu



Pingviini

Y = 1




A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

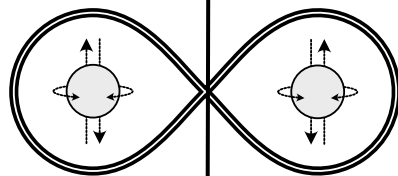
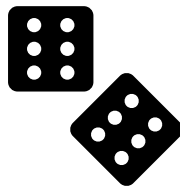
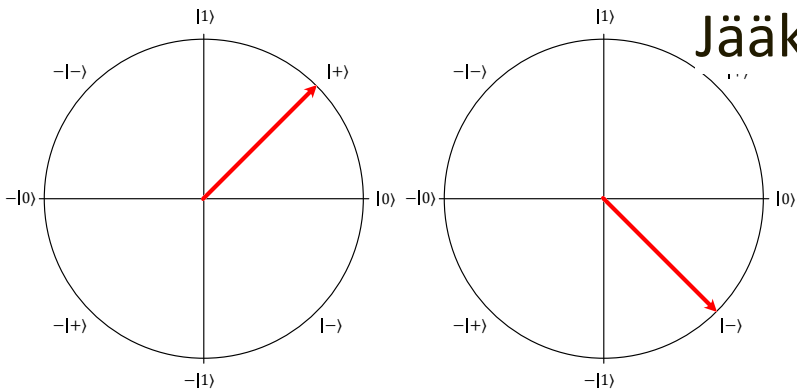
Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$ 0\rangle$	$\cos^2\left(\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(\frac{\pi}{8}\right) \approx 0.15$
1	$ 1\rangle$	$\cos^2\left(\frac{5\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{5\pi}{8}\right)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

CHSH game

X = 1

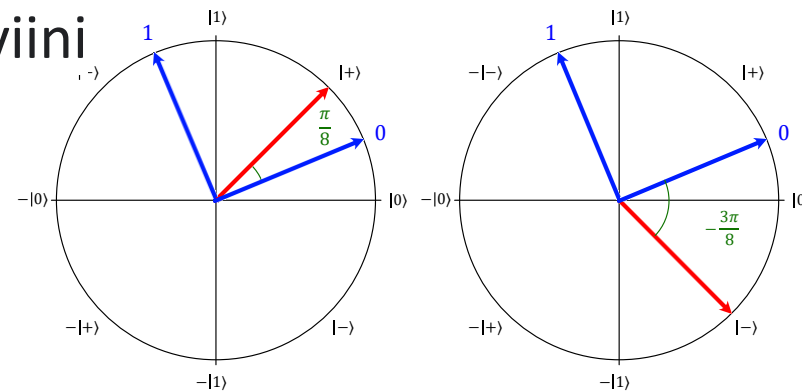



Jääkarhu



Pingviini

Y = 0



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

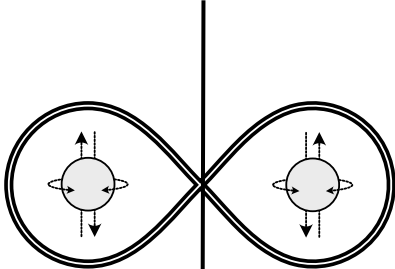
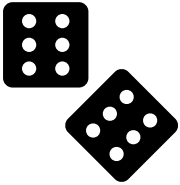
$X = 1$



X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1



Jääkarhu



Pingviini

$Y = 0$




A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

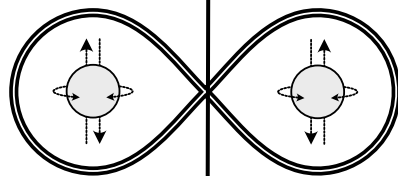
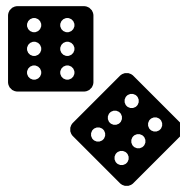
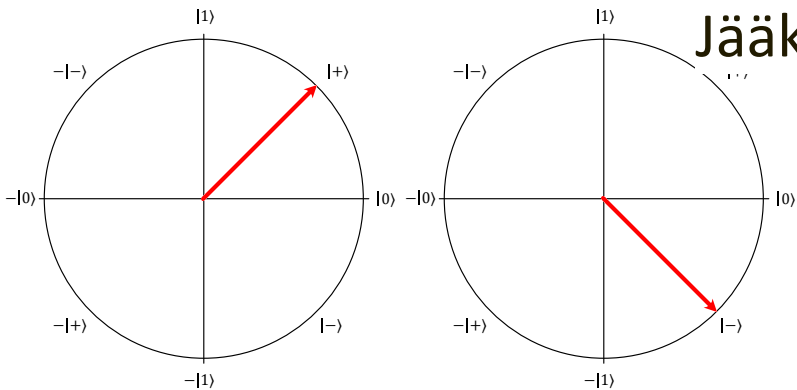
Jääkarhu outputs	Pingviini's qbit	Pingviini outputs 0 with probability	Pingviini outputs 1 with probability
0	$ +\rangle$	$\cos^2\left(\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(\frac{\pi}{8}\right) \approx 0.15$
1	$ -\rangle$	$\cos^2\left(-\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(-\frac{3\pi}{8}\right)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

CHSH game

X = 1

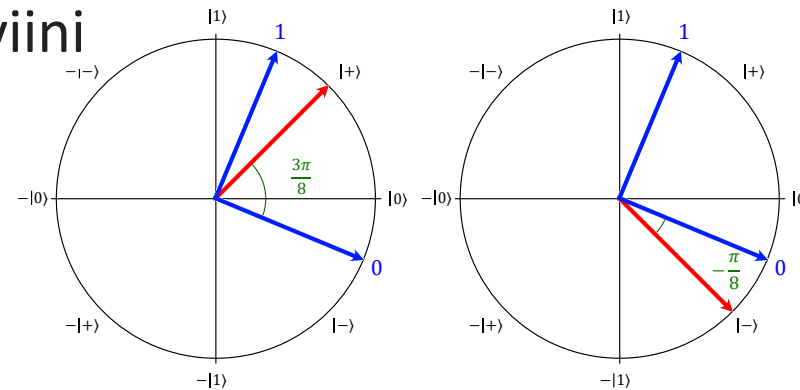



Jääkarhu



Pingviini

Y = 1



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

CHSH game

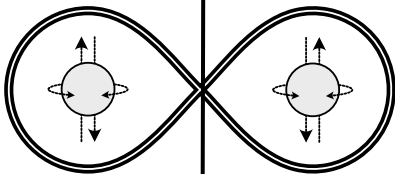
X = 1



X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1



Jääkarhu



Pingviini

Y = 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	$ +\rangle$	$\cos^2\left(\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{3\pi}{8}\right) \approx 0.85$
1	$ -\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

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

computing, but what does it mean to you as a software developer? With many new investment by some of the largest tech companies in the world to be the first to develop 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.

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

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

independent quantum tool set using the Microsoft Quantum Development Kit or Microsoft Visual Studio

mechanics required to start working on quantum computing

mathematical concepts such as complex numbers, trigonometry, and linear algebra

Microsoft Quantum Development Kit on a Windows or Linux PC with Visual Studio Code or Microsoft

Visual 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 Deutsch, Deutsch-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 Hooyberghs 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



9 781484 272459

Shelve In:
Microsoft

User level:
Beginning-Intermediate



Hooyberghs **Introducing Microsoft Quantum Computing for Developers**



Introducing Microsoft Quantum Computing for Developers

Using the Quantum Development Kit and Q#

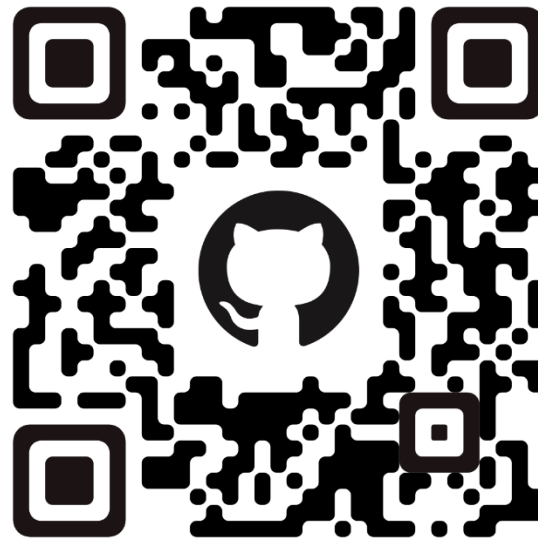
—
Johnny Hooyberghs

apress®
www.apress.com

apress®

apress®

johnny.hooyberghs@involved.be



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