Welsh Azure User Group

Azure Quantum & Microsoft Q#

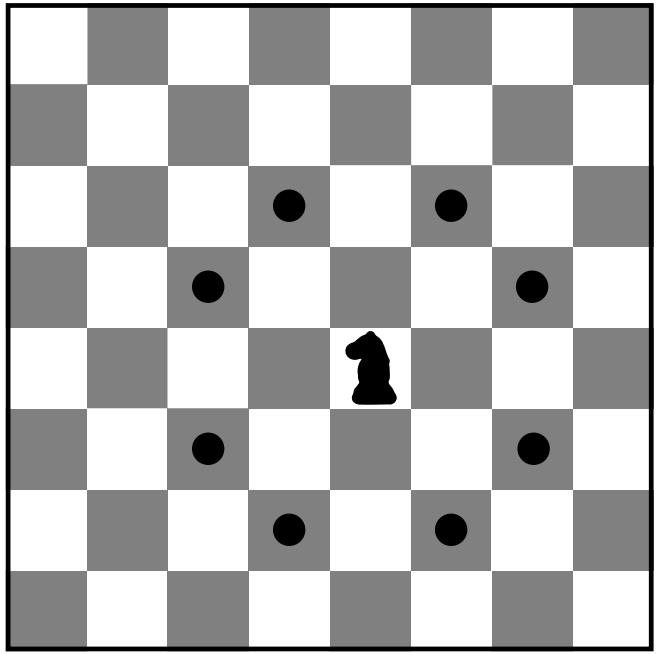
Johnny Hooyberghs

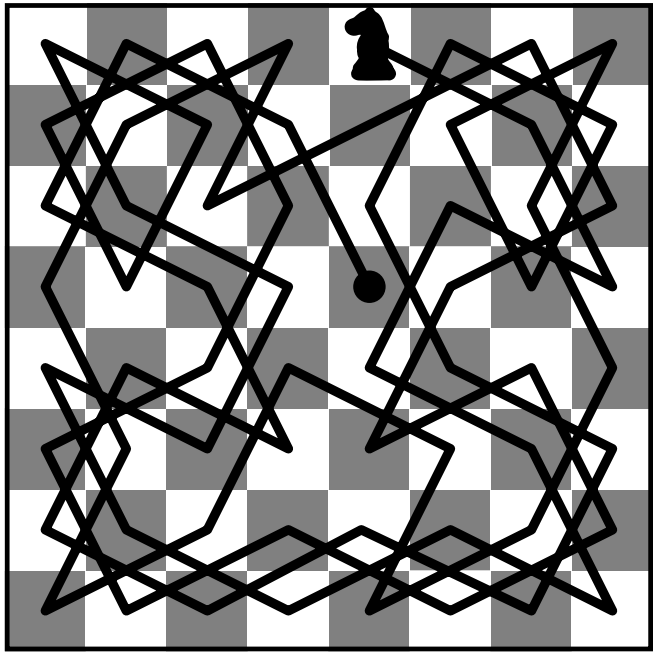
Here's Johnny

Johnny Hooyberghs
@djohnnieke
johnny.hooyberghs@involved.be

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

Why Quantum Computing?

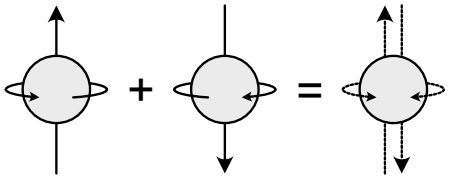


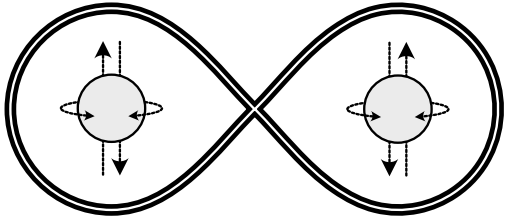


Why Quantum Computing?

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
- …

Bits vs. Qubits

0 1

Bits vs. Qubits

100110

Bits vs. Qubits

$$|0\rangle \quad |1\rangle$$

Bits vs. Qubits

$$|100110\rangle$$

Quantum State

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

Quantum State

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

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

Quantum State

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

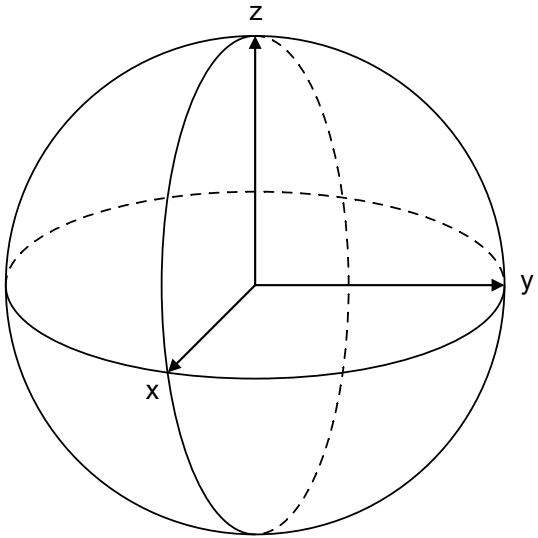
$$|\alpha|^2 + |\beta|^2 = 1$$

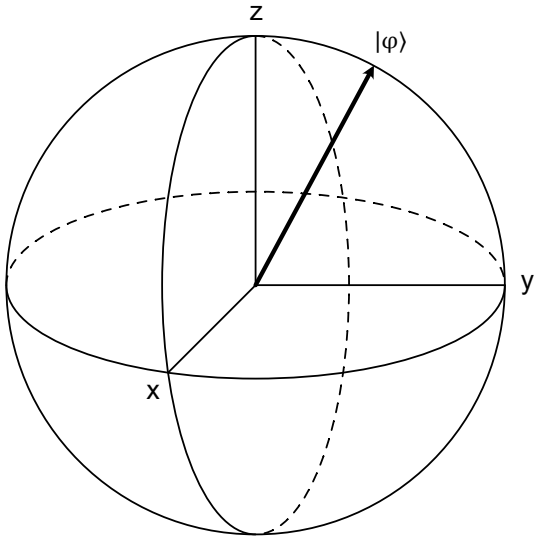
$$\alpha = a + bi \qquad \beta = c + di$$

Quantum State

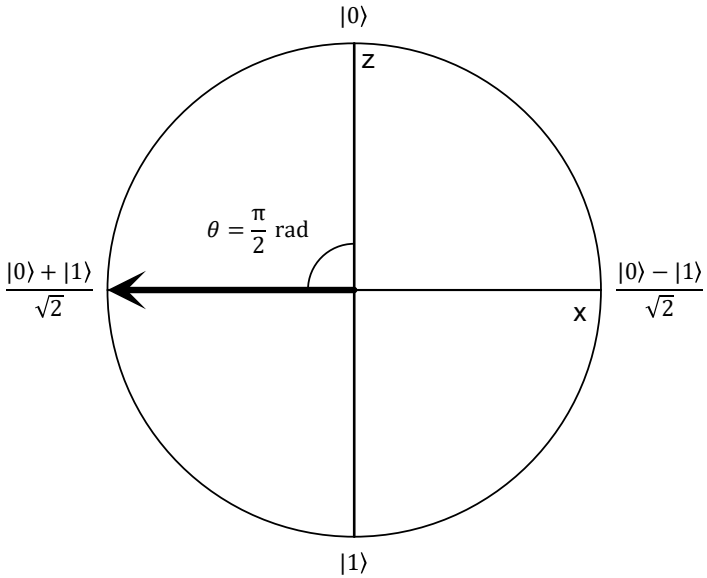
$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

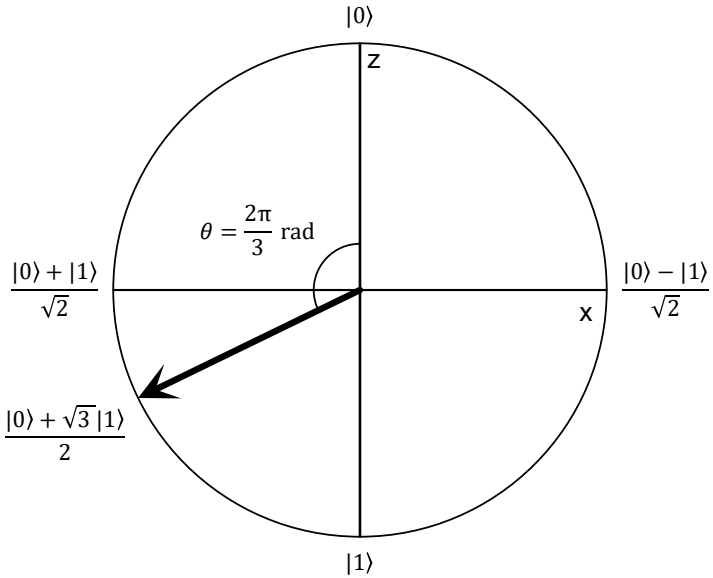
Quantum State





Quantum State





Quantum State

2 Qubit system (4 probabilities):

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

Quantum State

2 Qubit system (4 probabilities):

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

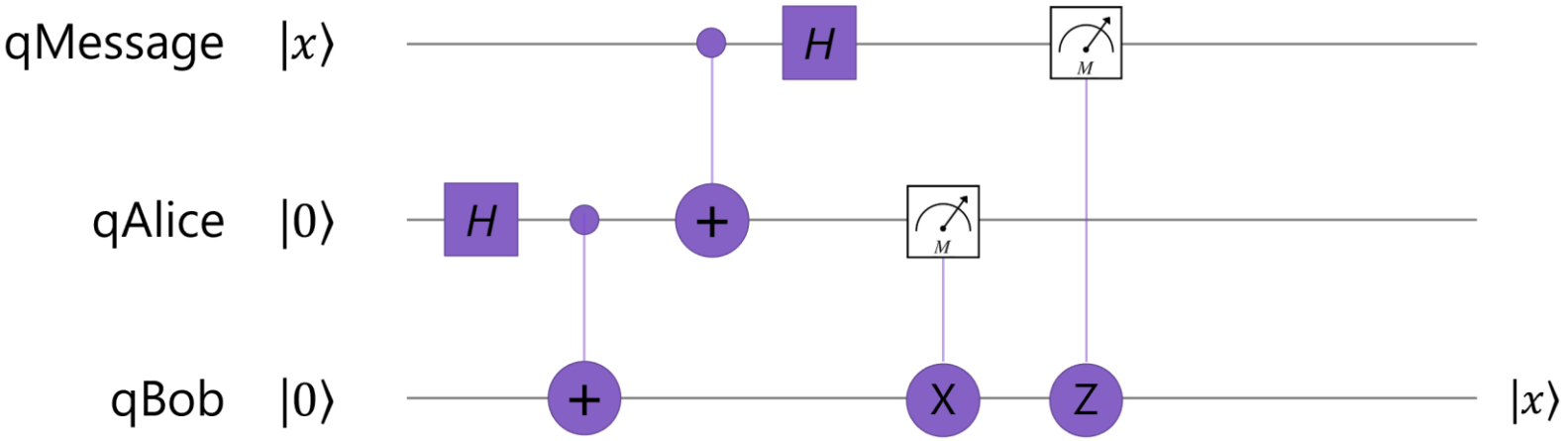
3 Qubit system (8 probabilities):

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

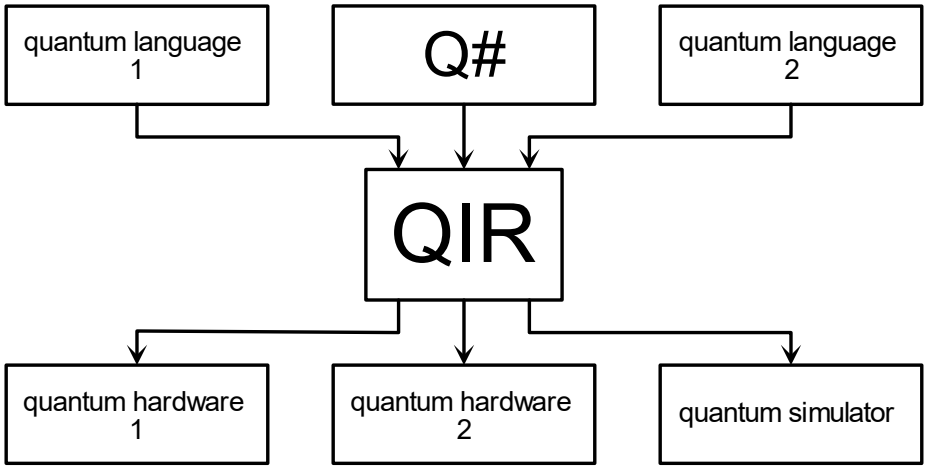
4 Qubit system (16 probabilities):

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

Teleportation



Azure Quantum

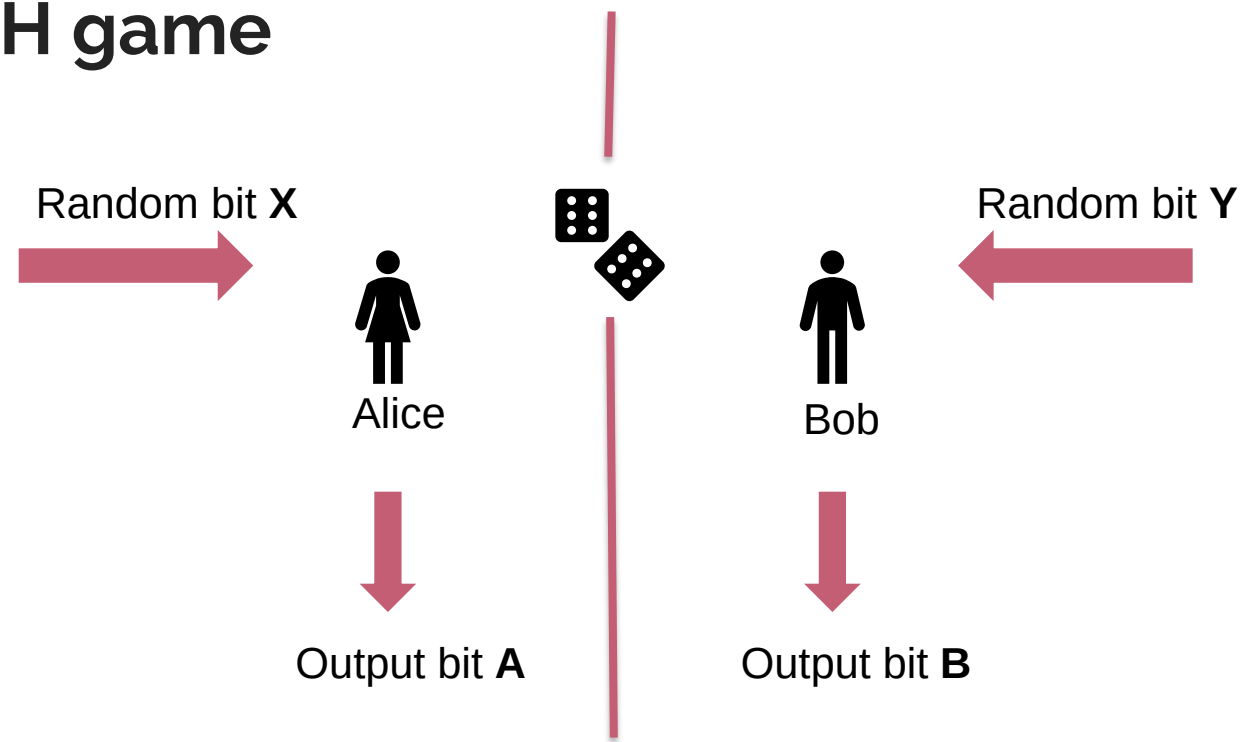


Quantum in the cloud
- Optimization
- Machine Learning
- Quantum Simulation

Access to quantum hardware
- Microsoft (Topological)
- IonQ & Quantinuum (Ion Traps)
- QCI (Superconducting)

Q# & QDK
- Quantum Intermediate Representation (QIR)

CHSH game



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

CHSH game

Alice

Bob

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

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

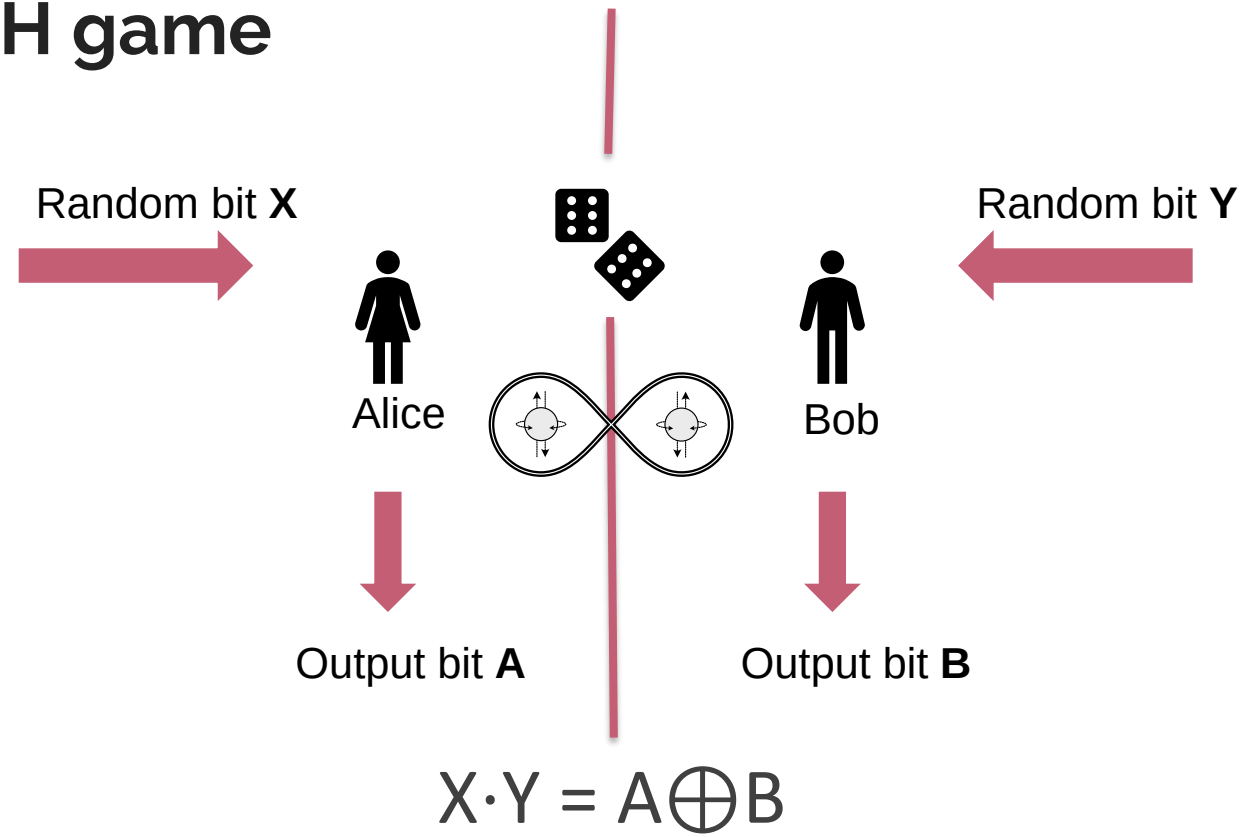
CHSH game

X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

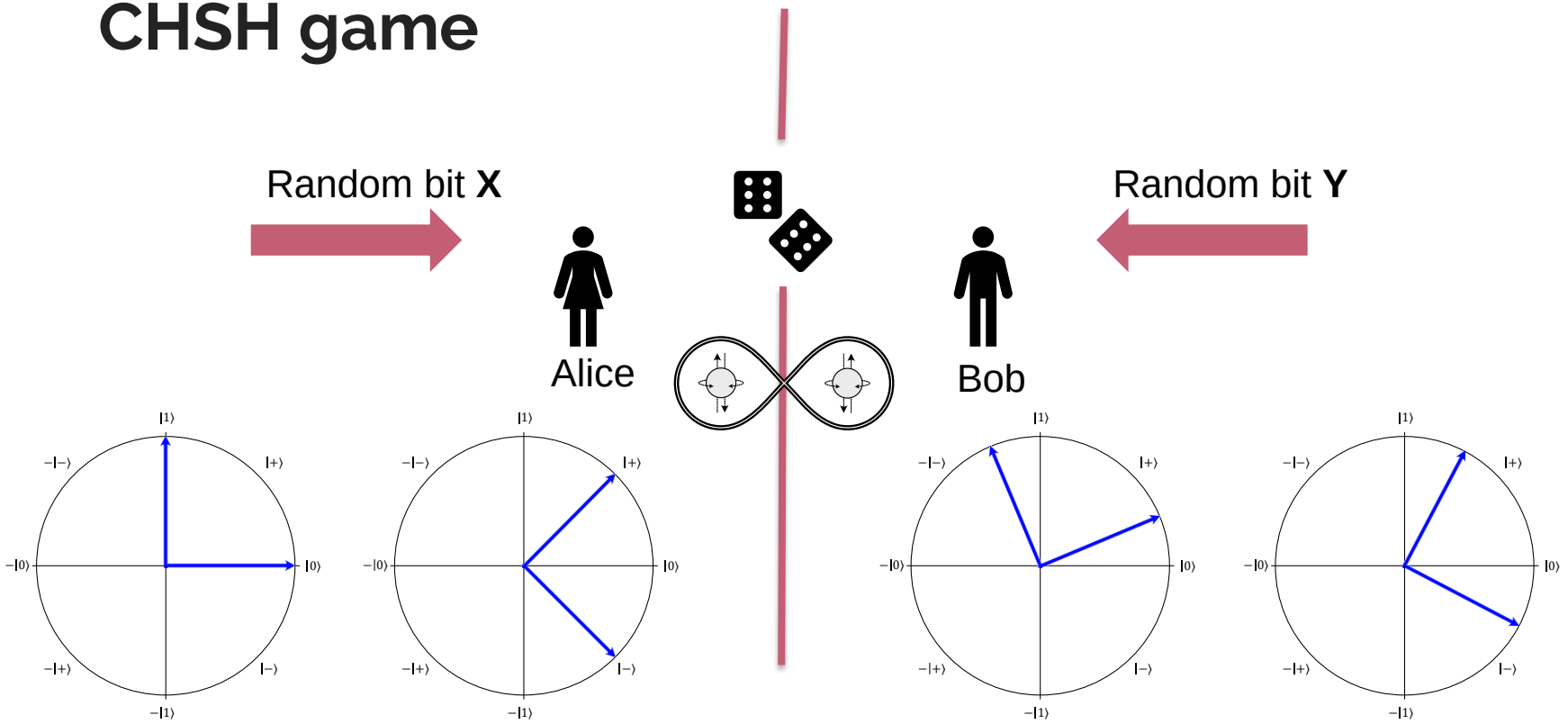
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

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

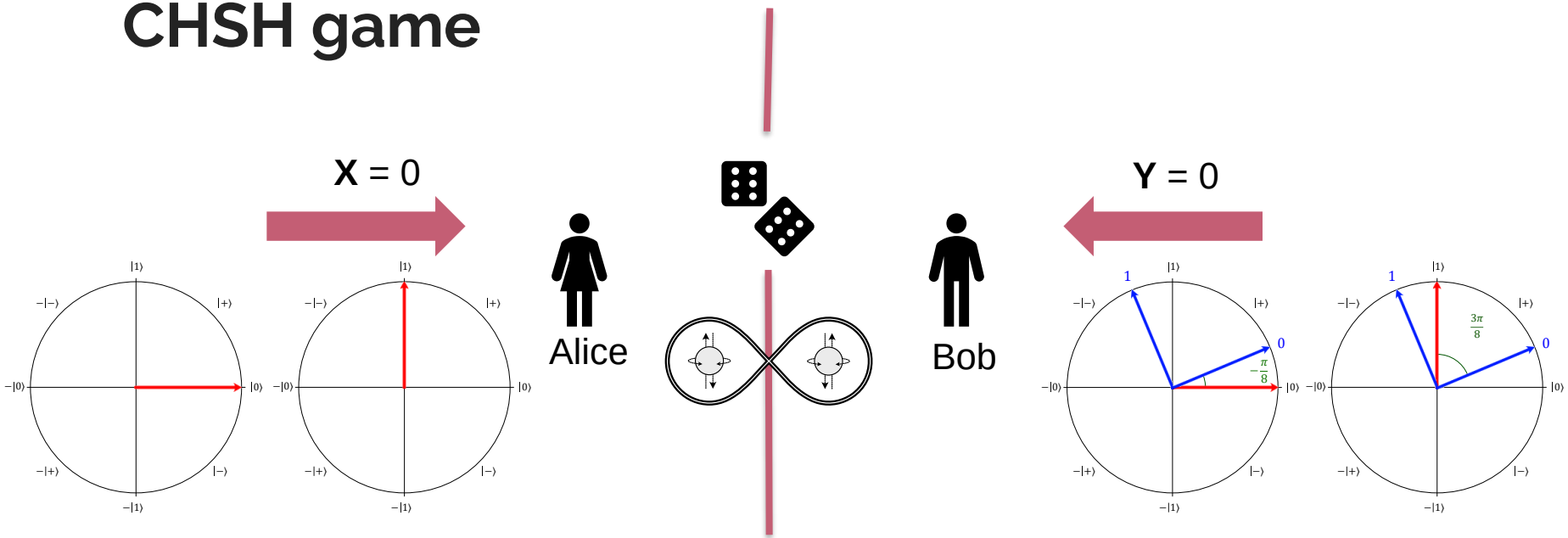
CHSH game


Random bit **X**
Random bit **Y**
Alice
Bob
Output bit **A**
Output bit **B**
X·Y = A⊕B

CHSH game

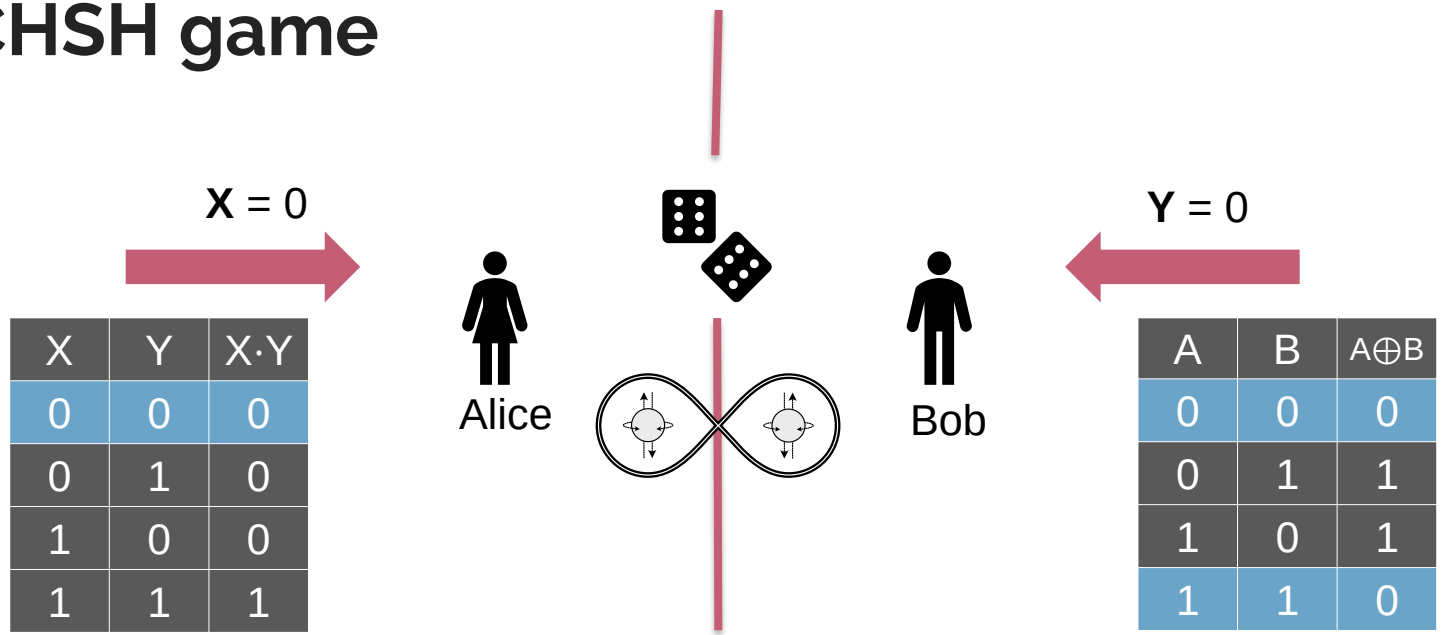


CHSH game



Alice outputs	Bob's qbit	Bob outputs 0 with probability	Bob 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

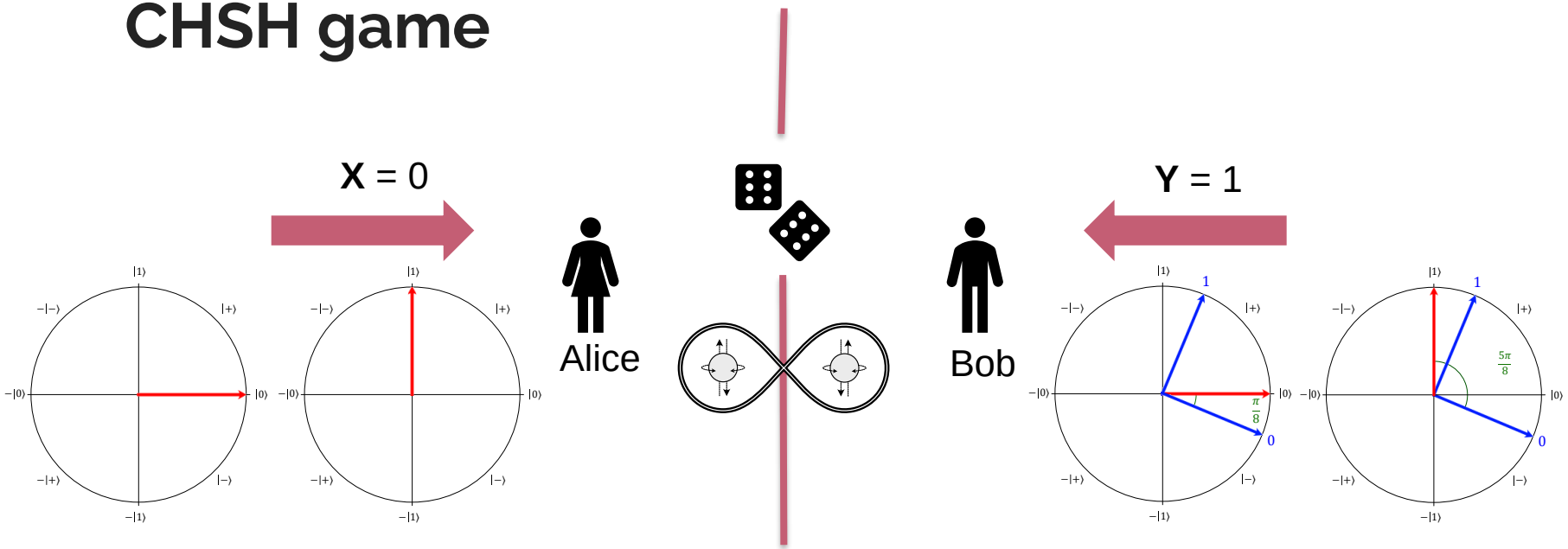


X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

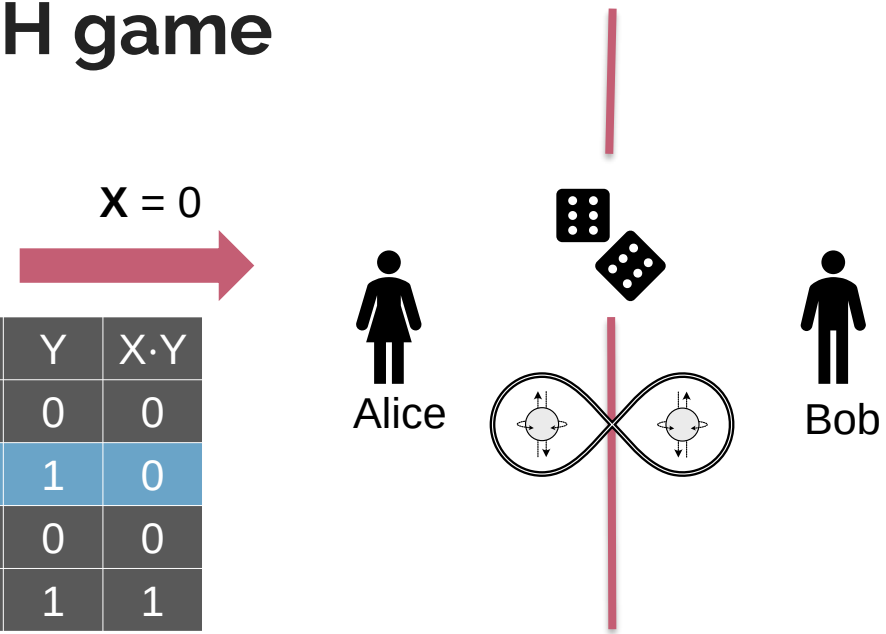
Alice outputs	Bob's qbit	Bob outputs 0 with probability	Bob outputs 1 with probability
0	$\|0\rangle$	$\cos^2\left(-\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(-\frac{\pi}{8}\right) \approx 0.15$
1	$\|1\rangle$	$\cos^2\left(\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{3\pi}{8}\right) \approx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

CHSH game



Alice outputs	Bob's qbit	Bob outputs 0 with probability	Bob 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

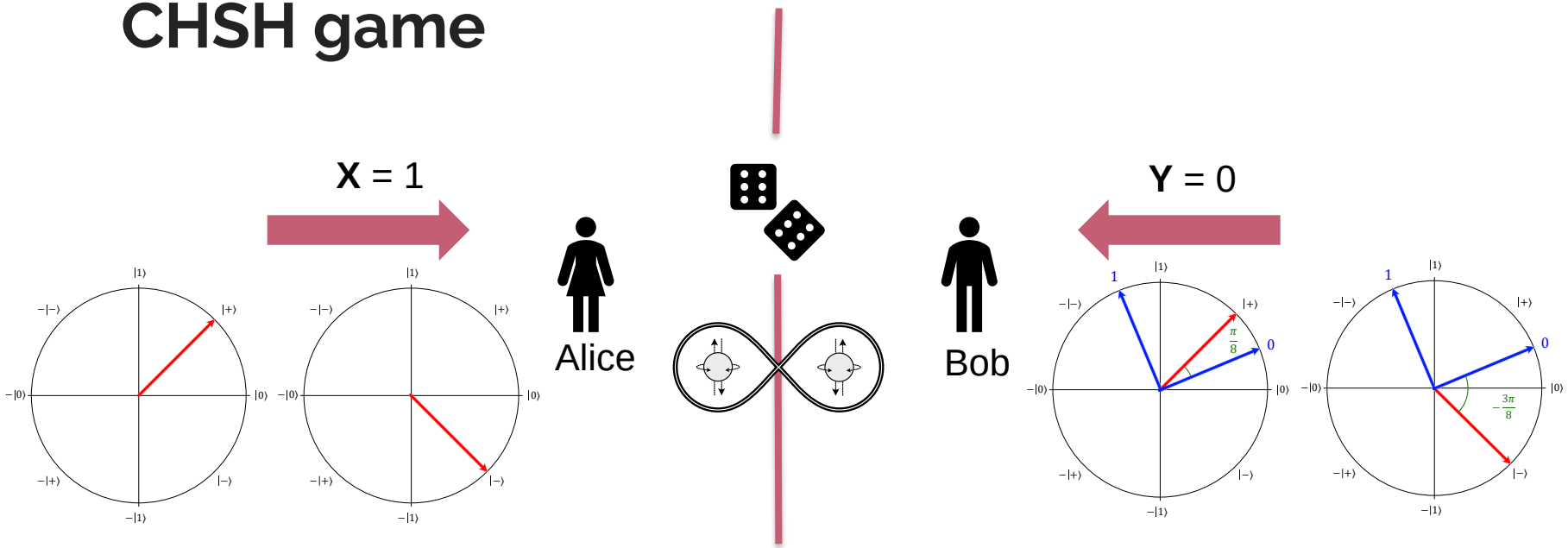


X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

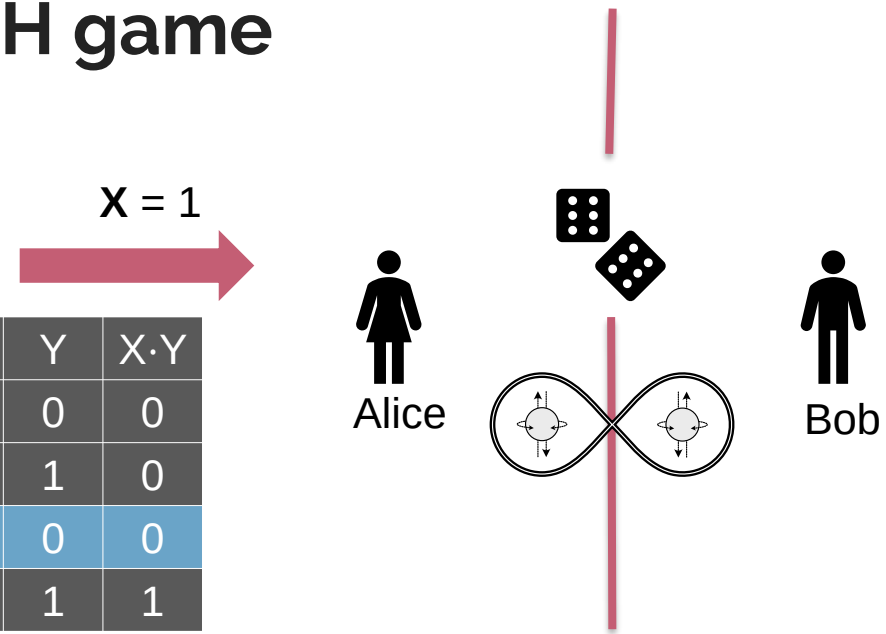
Alice outputs	Bob's qbit	Bob outputs 0 with probability	Bob outputs 1 with probability
0	$\|0\rangle$	$\cos^2\left(\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(\frac{\pi}{8}\right) \approx 0.15$
1	$\|1\rangle$	$\cos^2\left(\frac{5\pi}{8}\right) \approx 0.15$	$\sin^2\left(\frac{5\pi}{8}\right) \approx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

CHSH game



Alice outputs	Bob's qbit	Bob outputs 0 with probability	Bob 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

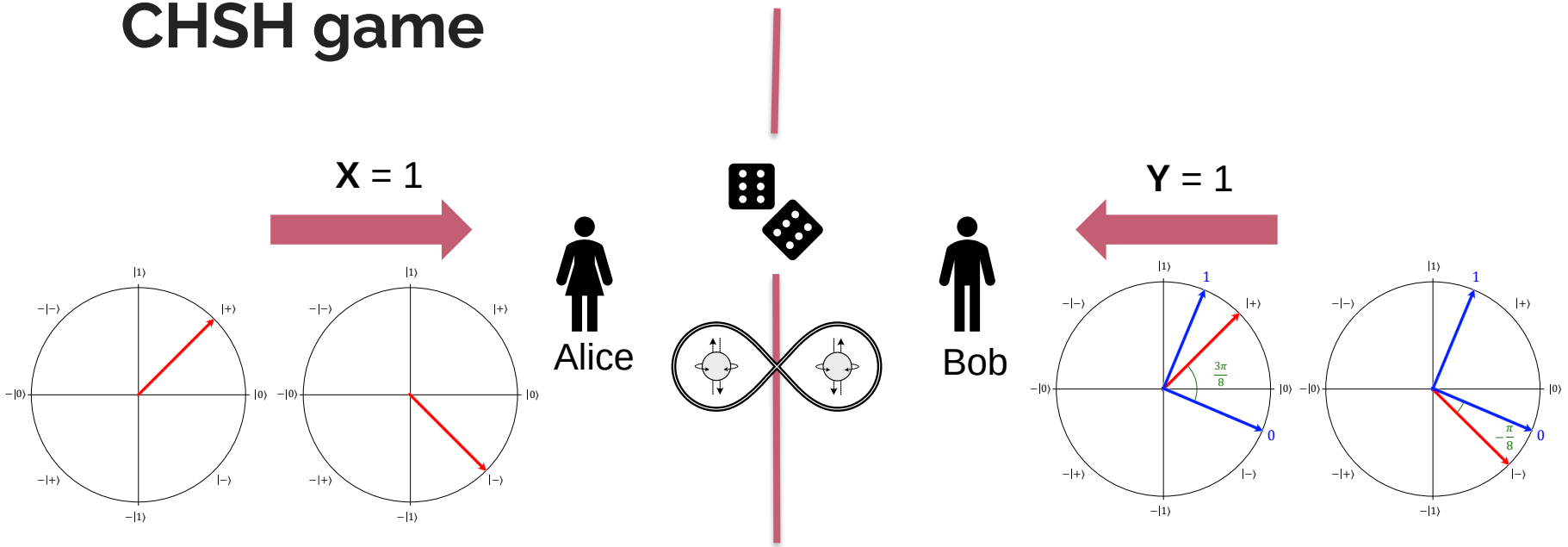


X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

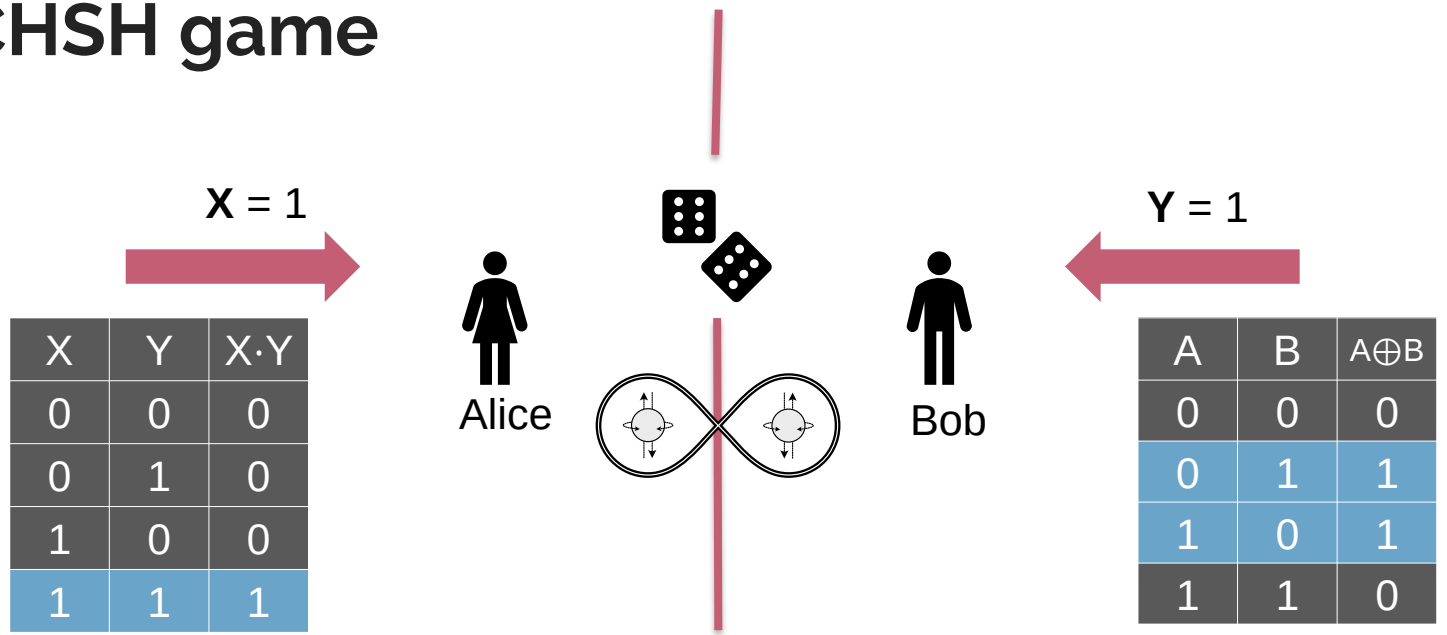
Alice outputs	Bob's qbit	Bob outputs 0 with probability	Bob outputs 1 with probability
0	$\|+\rangle$	$\cos^2\left(\frac{\pi}{8}\right) \approx 0.85$	$\sin^2\left(\frac{\pi}{8}\right) \approx 0.15$
1	$\|-\rangle$	$\cos^2\left(-\frac{3\pi}{8}\right) \approx 0.15$	$\sin^2\left(-\frac{3\pi}{8}\right) \approx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

CHSH game



Alice outputs	Bob's qbit	Bob outputs 0 with probability	Bob 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	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

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

Introducing Microsoft Quantum Computing for Developers

Dive in with this hands-on introduction to quantum computing with the Microsoft Quantum Development Kit and Q# for software developers.

You may have heard about quantum computing, but what does it mean to you as a software developer? With many new developments, a resurgence of interest, and investment by some of the largest tech companies in the world to be the first to market with quantum programming (QP) hardware and platforms, it is no longer a tool in the distant future. Developers are at the forefront, now able to create applications that take advantage of QP through simulations. While the skill is of interest, for many developers, quantum computing and its implications still remains a mystery.

In this hands-on book, you will get up to speed exploring important quantum concepts and apply them in practice through writing actual quantum algorithms, using the Microsoft Quantum Development Kit. Theoretical knowledge about quantum physics, such as superposition and entanglement, will be used to explain quantum computing topics, including quantum gates, quantum circuits, and quantum algorithms. Finally, take a tour of the new Azure Quantum.

Use Q#, Microsoft's new programming language, to target quantum hardware. You will select your supporting language of choice, either C# or Python, to begin writing your quantum applications. Combined with just enough theoretical preparation, you will learn how to get your computer ready to simulate basic quantum programs using Microsoft Visual Studio or Visual Studio Code and Q#.

What You Will Learn

- Get up to speed on the platform-independent quantum tool set using the Microsoft Quantum Development Kit simulator and Visual Studio Code or Microsoft Visual Studio
- Know the basics of quantum mechanics required to start working on quantum computing
- Understand mathematical concepts such as complex numbers, trigonometry, and linear algebra
- Install the 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 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 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
54999
9 781484 272459

Shelve in:
Microsoft

User level:
Beginning–Intermediate

Microsoft Most Valuable Professional

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®

Questions?

Johnny Hooyberghs
@djohnnieke
johnny.hooyberghs@involved.be