



Fast Pattern Matching in Quantum Circuits

Luca Mondada & Pablo Andres-Martinez

IWQC 2023, Paris France, 21st July 2023

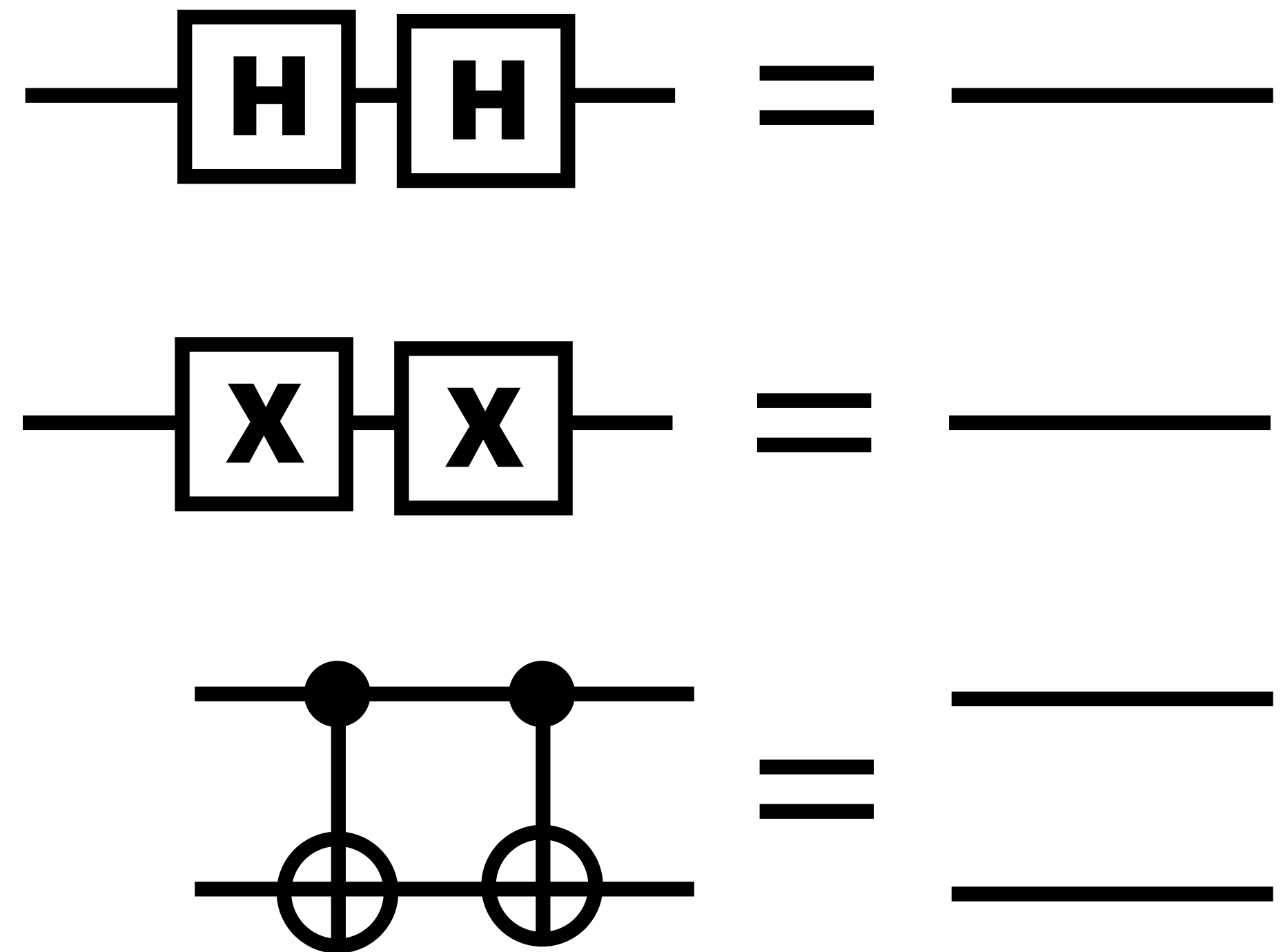
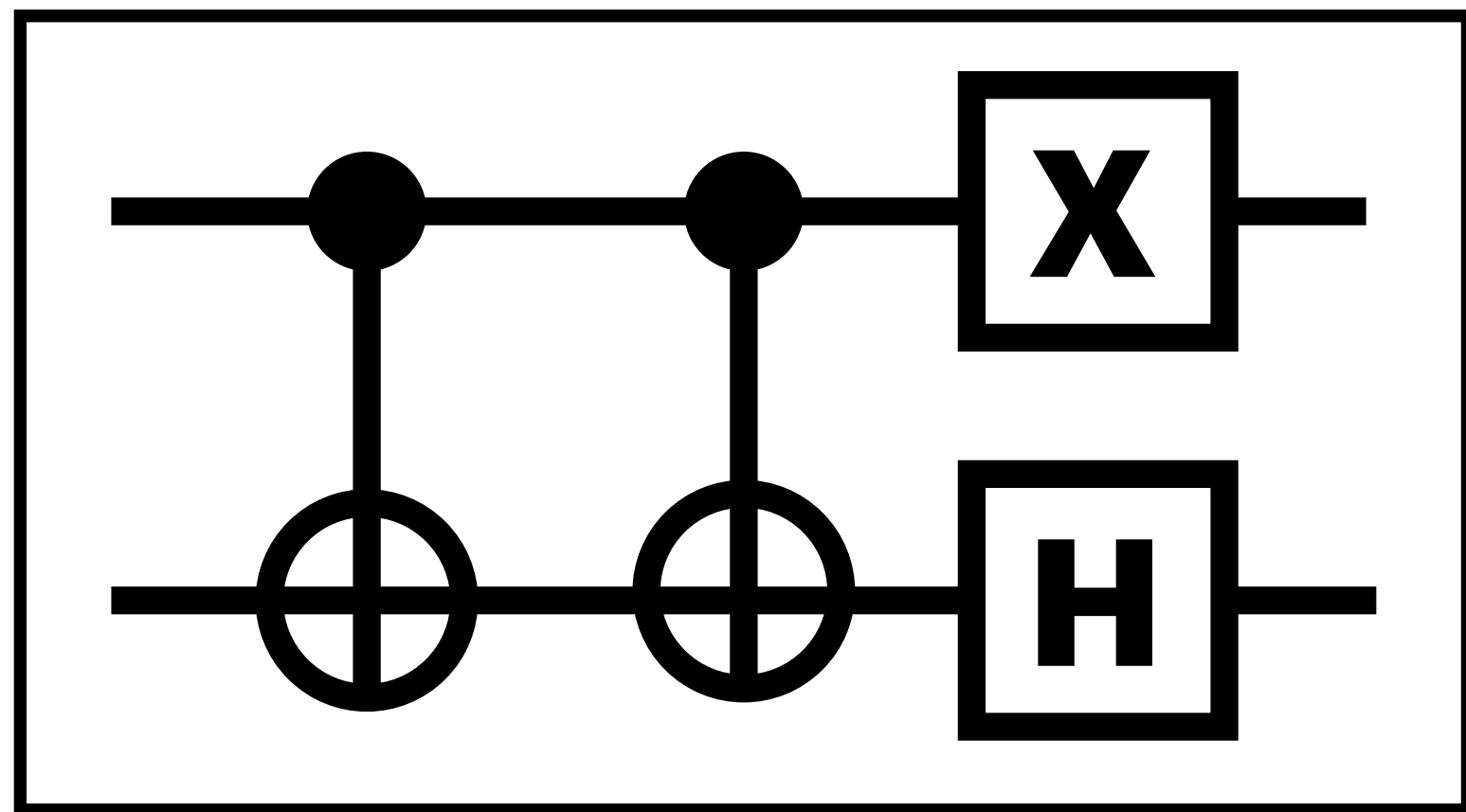
[arXiv:2302.06717](https://arxiv.org/abs/2302.06717) [quant-ph]

Quantum Circuit optimisation

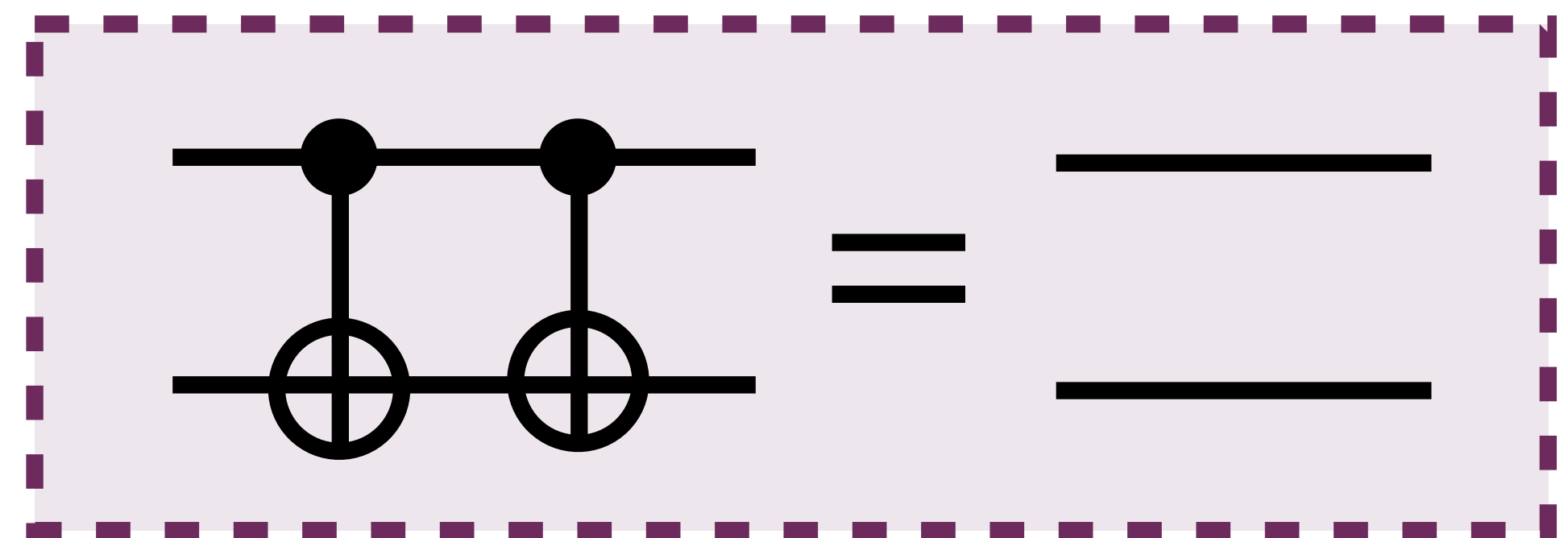
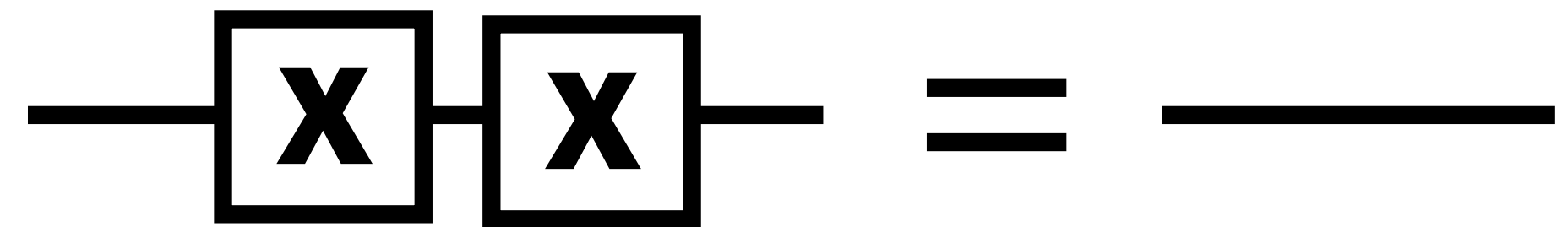
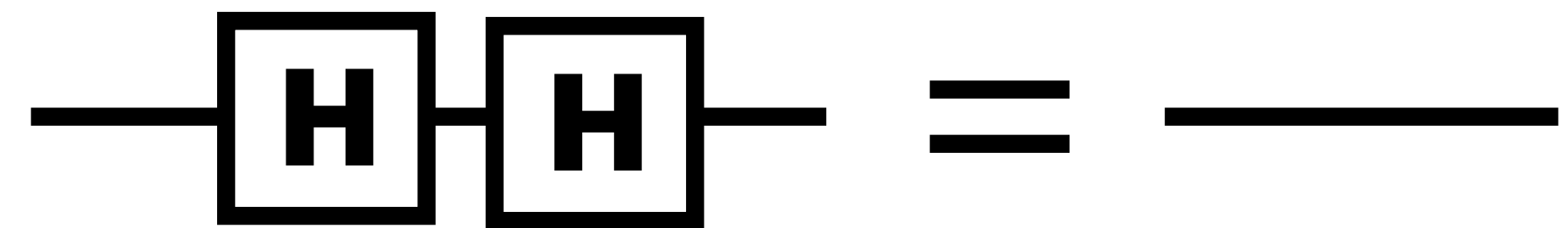
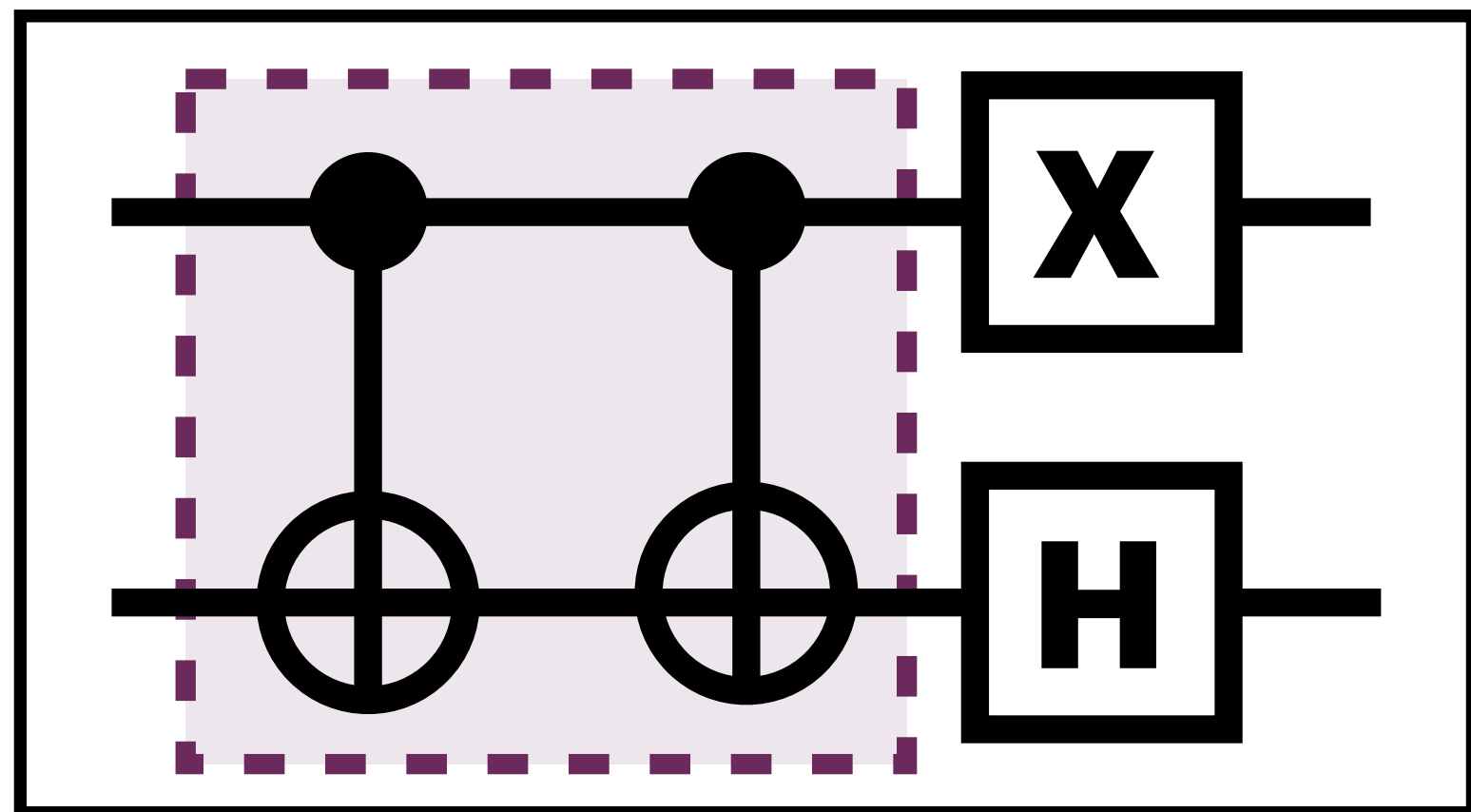
...is about to get a lot harder

1. Larger circuits
2. Hybrid classical-quantum computations
3. Every bit of optimisation will matter

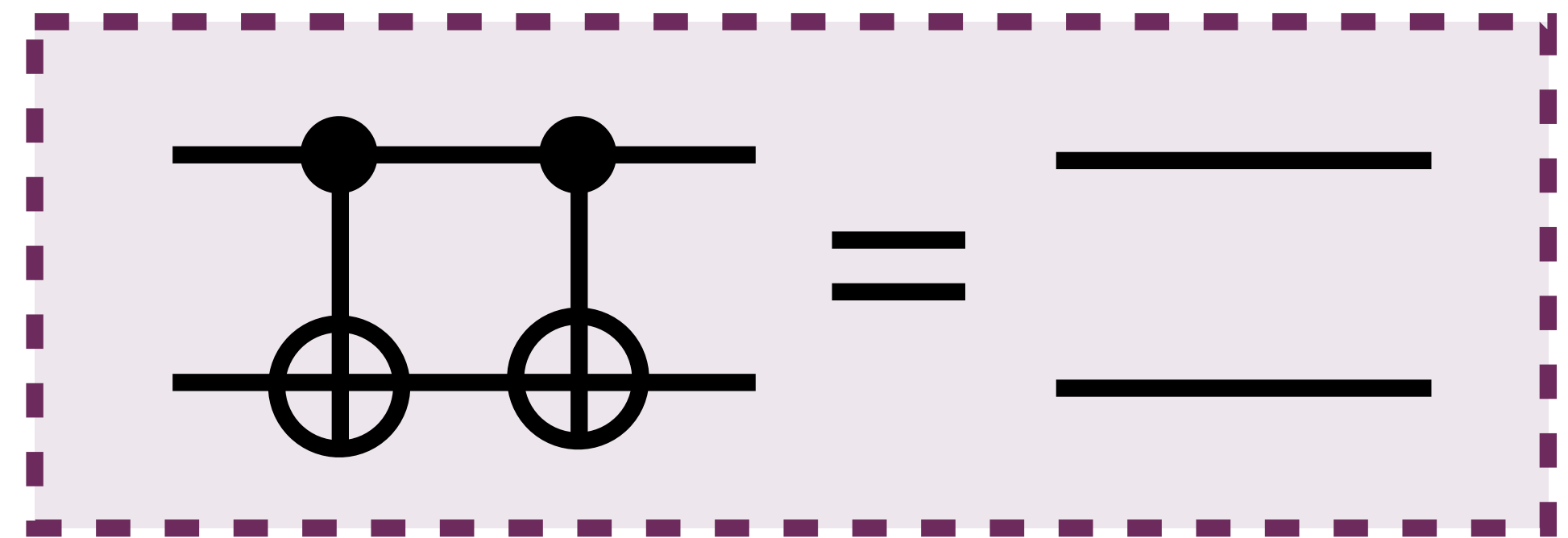
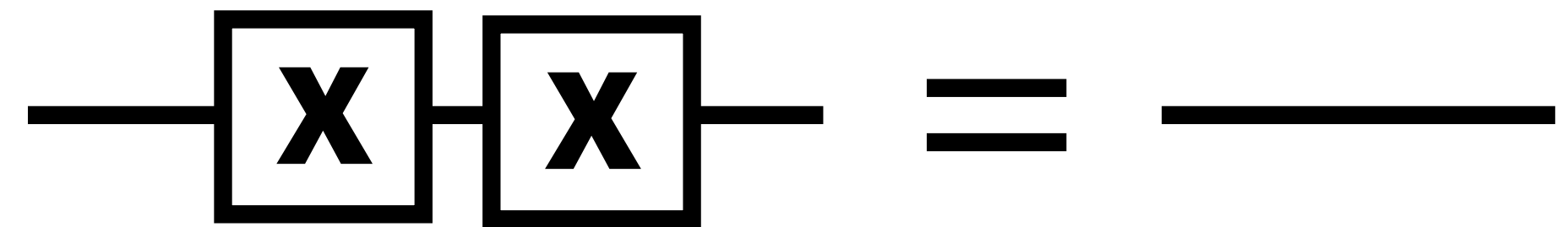
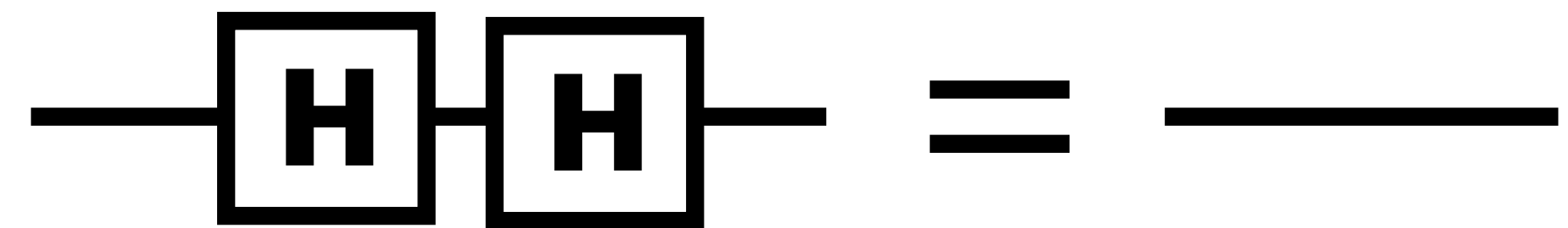
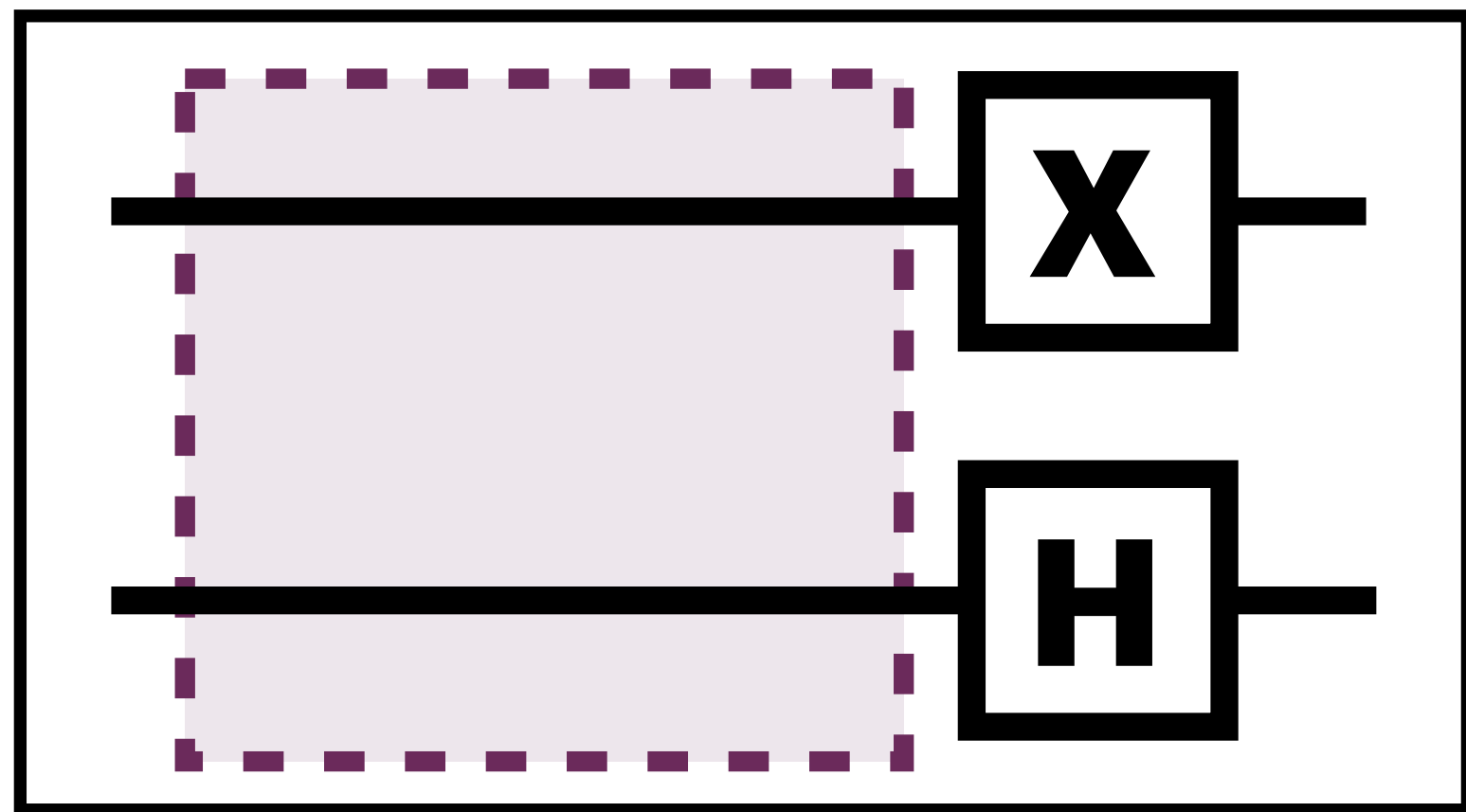
The simplest circuit optimisation framework




The simplest circuit optimisation framework




The simplest circuit optimisation framework



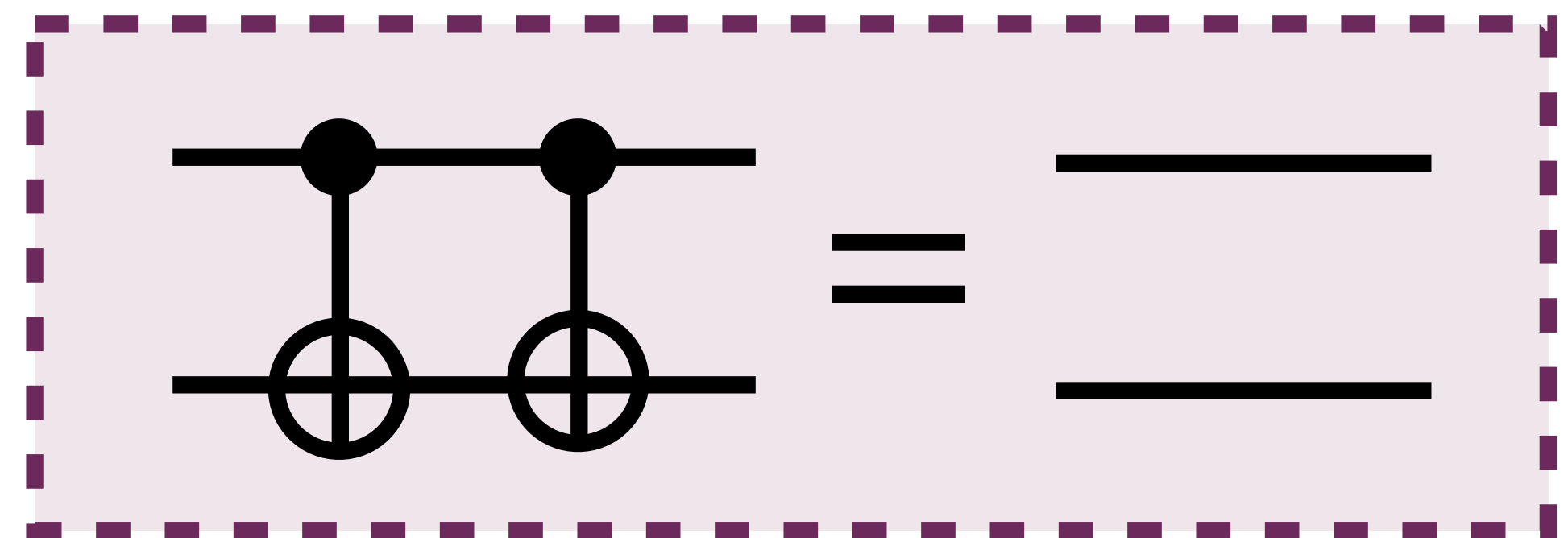
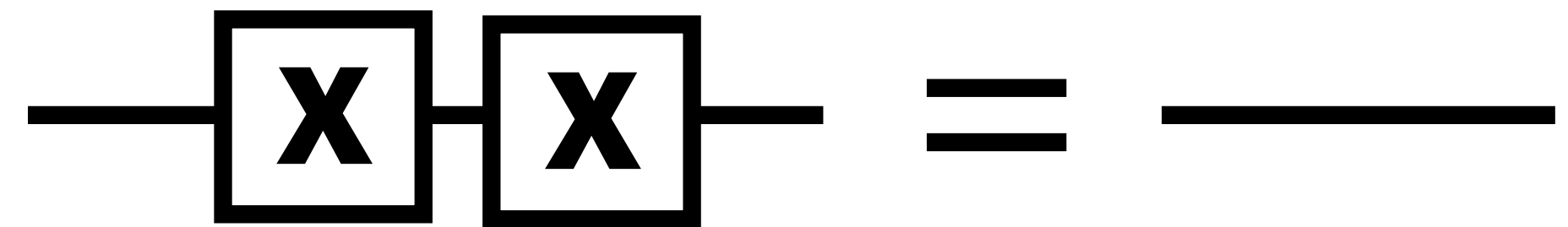
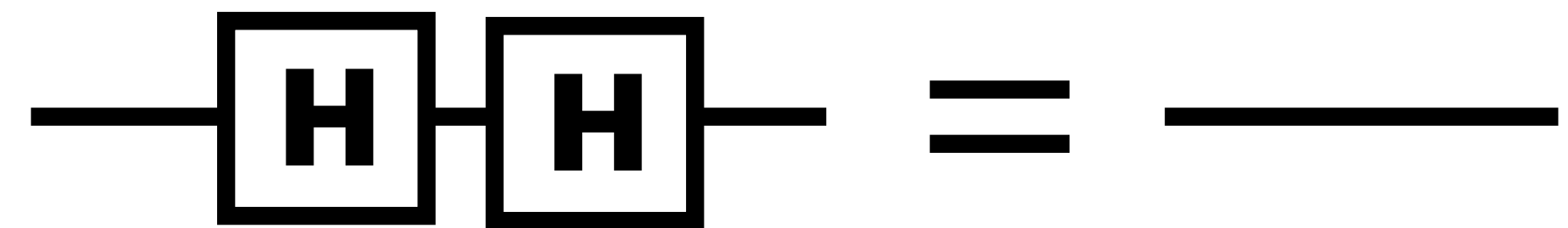
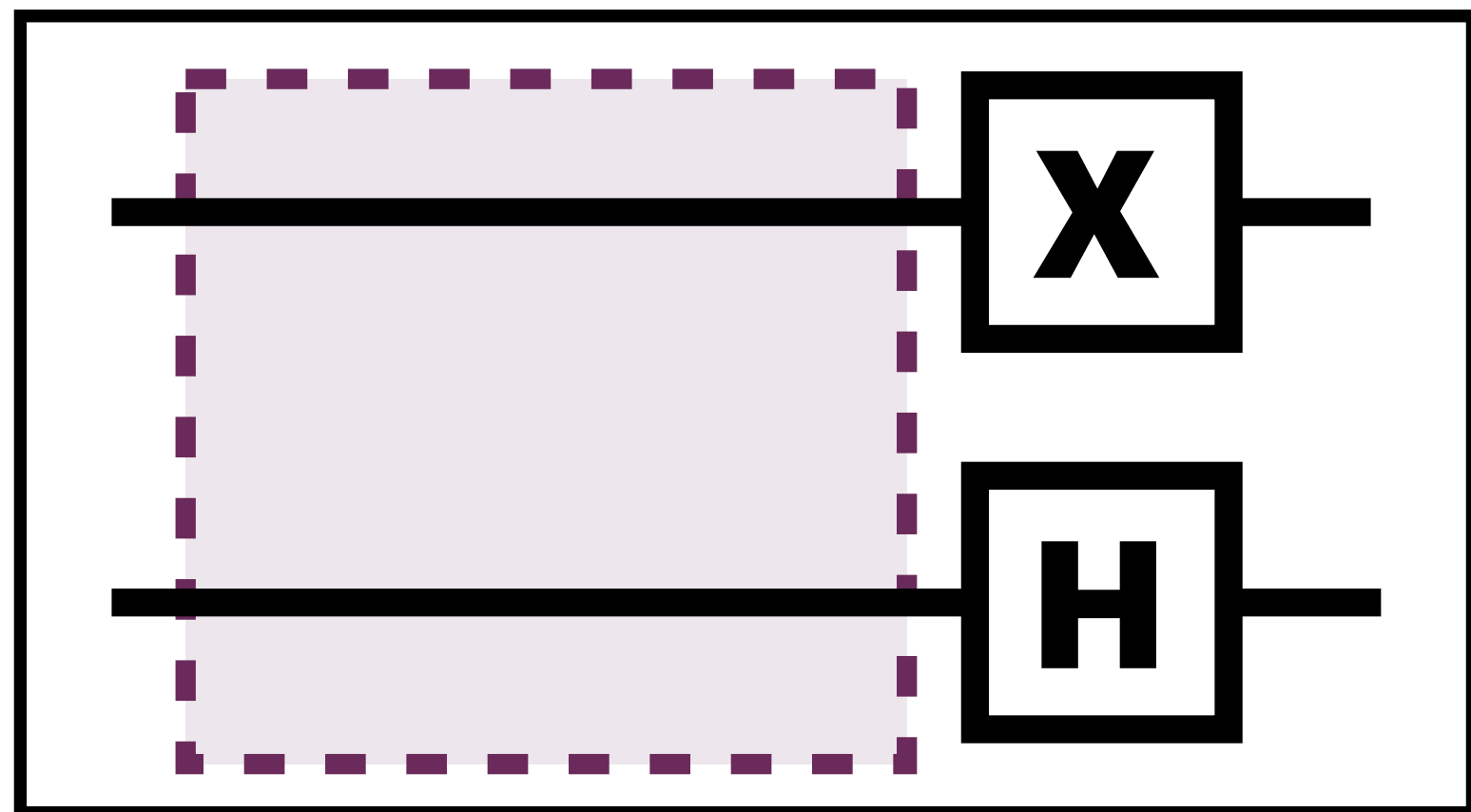
The simplest circuit optimisation framework

1. **Generate rewrite rules** (offline)
 2. **Find patterns in input**
 3. **Rewrite & optimise**
- 

The simplest circuit optimisation framework

1. **Generate rewrite rules** (offline)
 2. **Find patterns in input**
 3. **Rewrite & optimise**
- 

The simplest circuit optimisation framework




The simplest circuit optimisation framework



Solution: L-independent online runtime

Pattern matching complexity for strings VS circuits VS graphs

	EASY			HARD	
	strings		circuits		graphs
1 PATTERN	$O(n)$	[KMP]	$O(n \cdot m)$		\mathcal{NP} -complete
L PATTERNS	$O(n)$	[AC]	$O(n \cdot 2^q \cdot w)$		$O(n^2)$
precomputation:	$O(Lm)$		$O(Lm + w^q)$		$O(L3^m)$

this work 

WIP

n: size of input

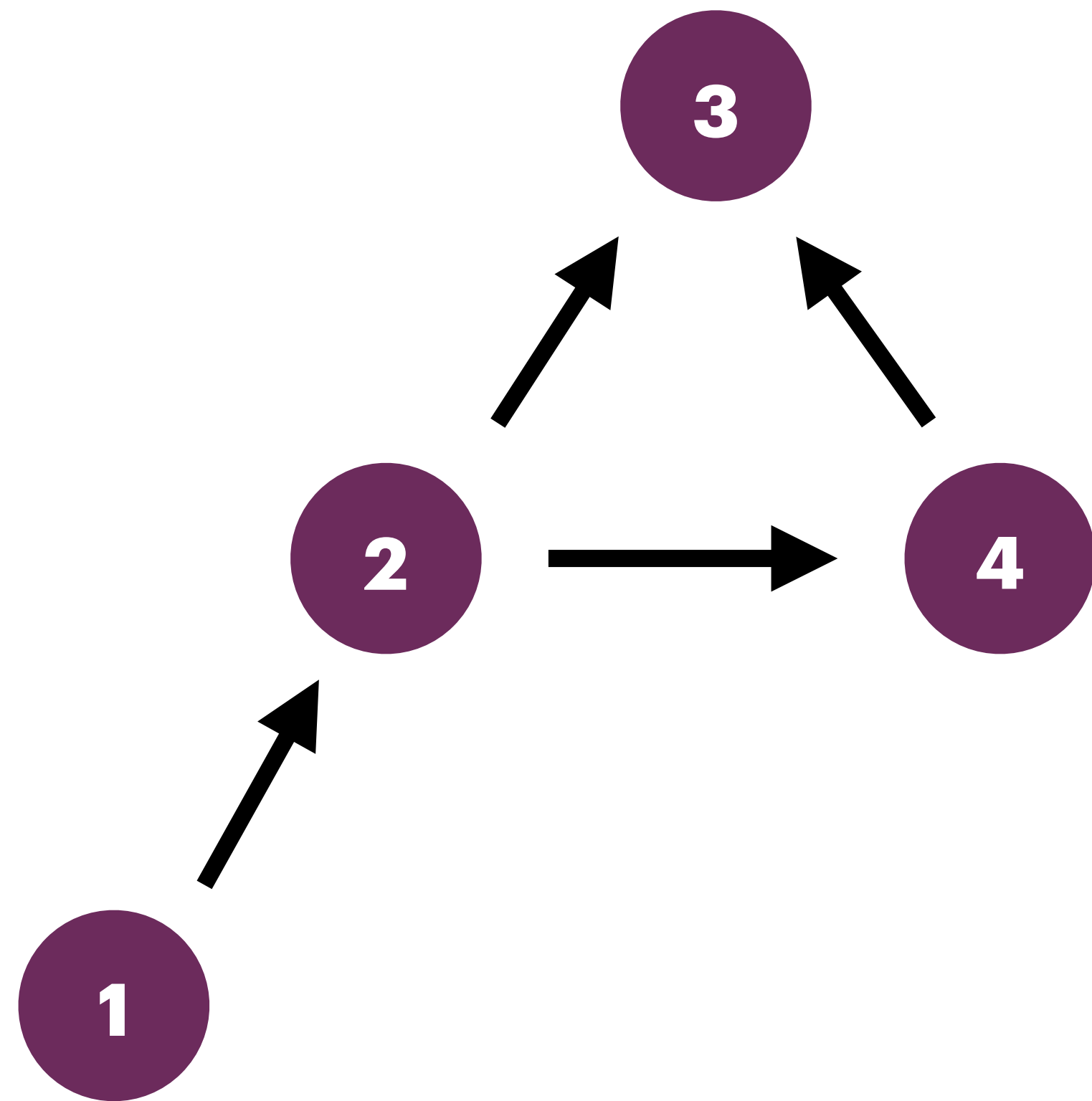
w: number of gates on one qubit in patterns

m: size of patterns

L: number of patterns
for circuits: $m = w \cdot q$

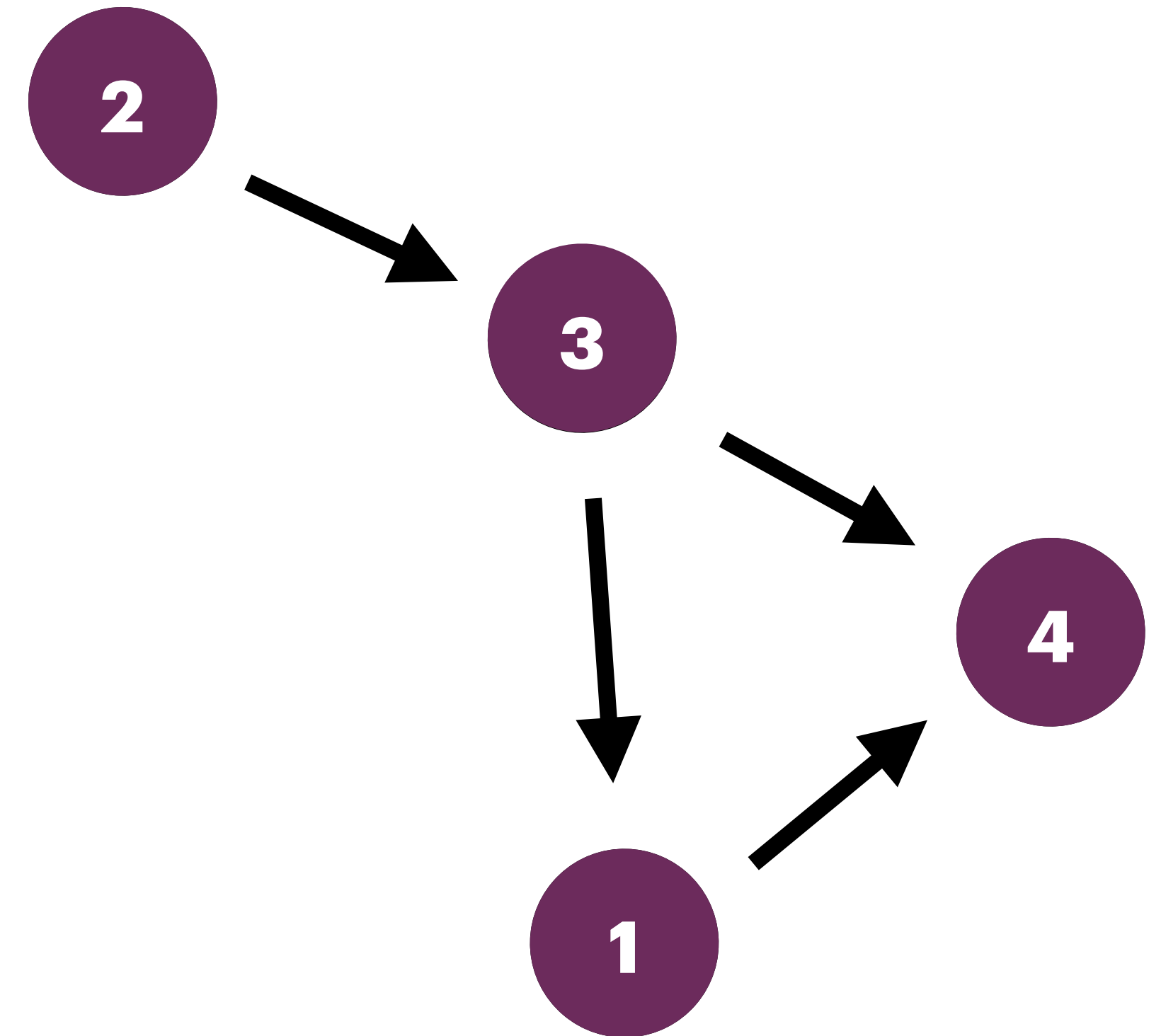
q: number of qubits in patterns

It's all about the vertex labelling

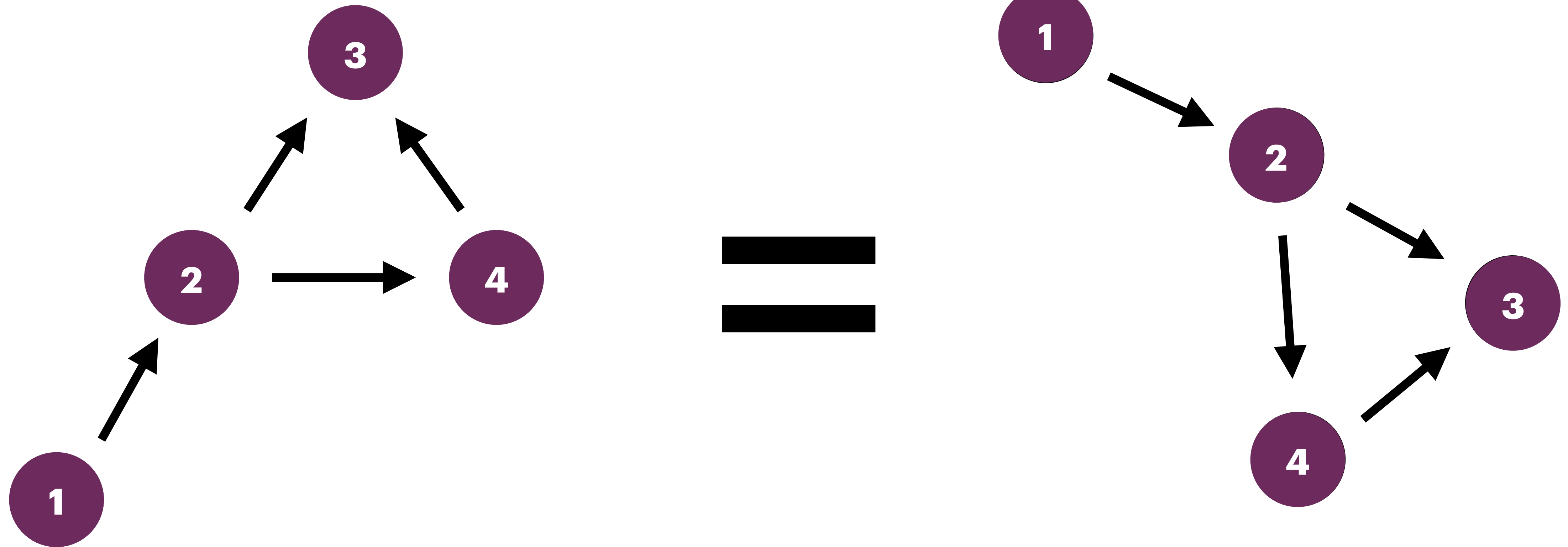


?

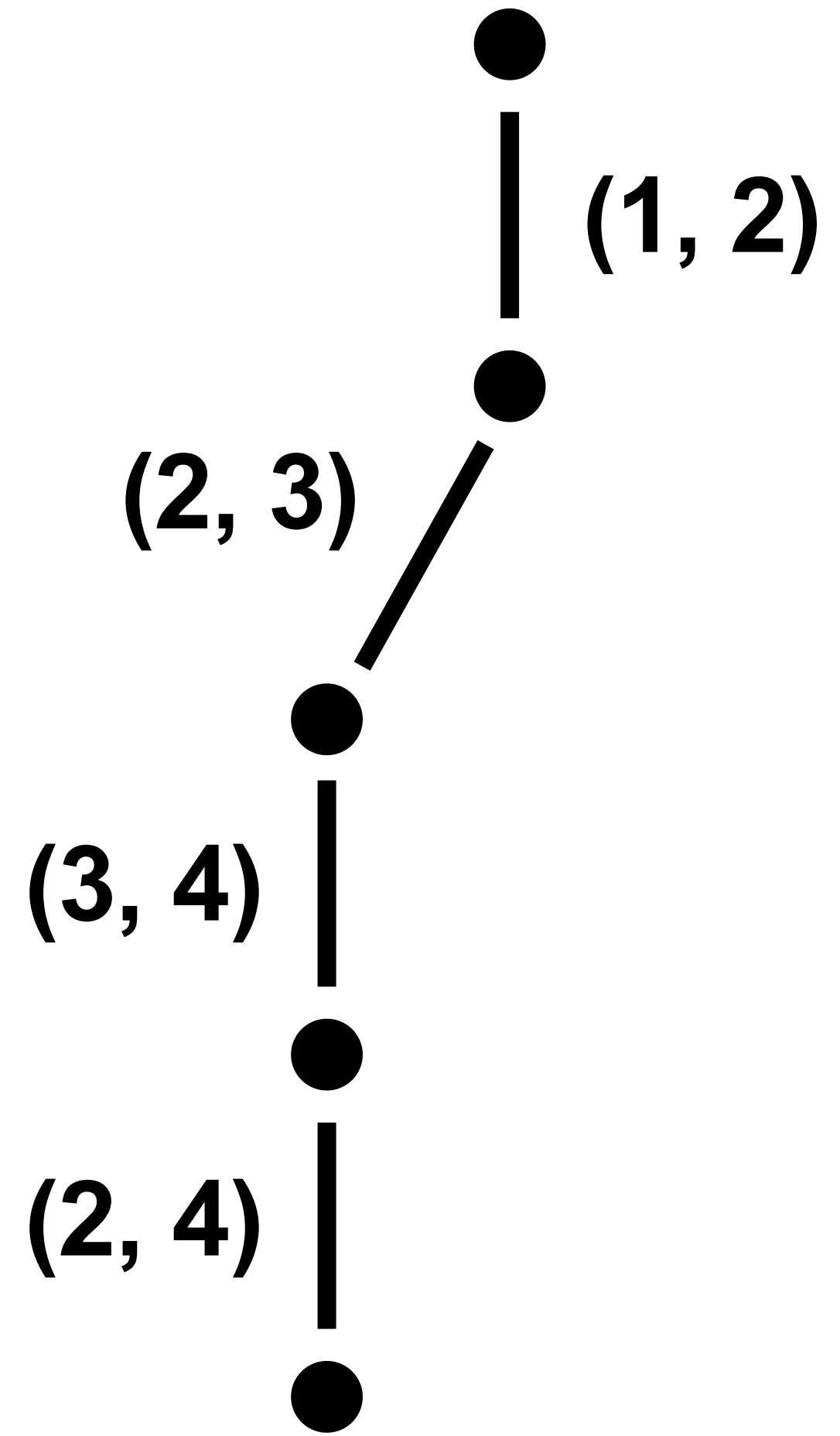
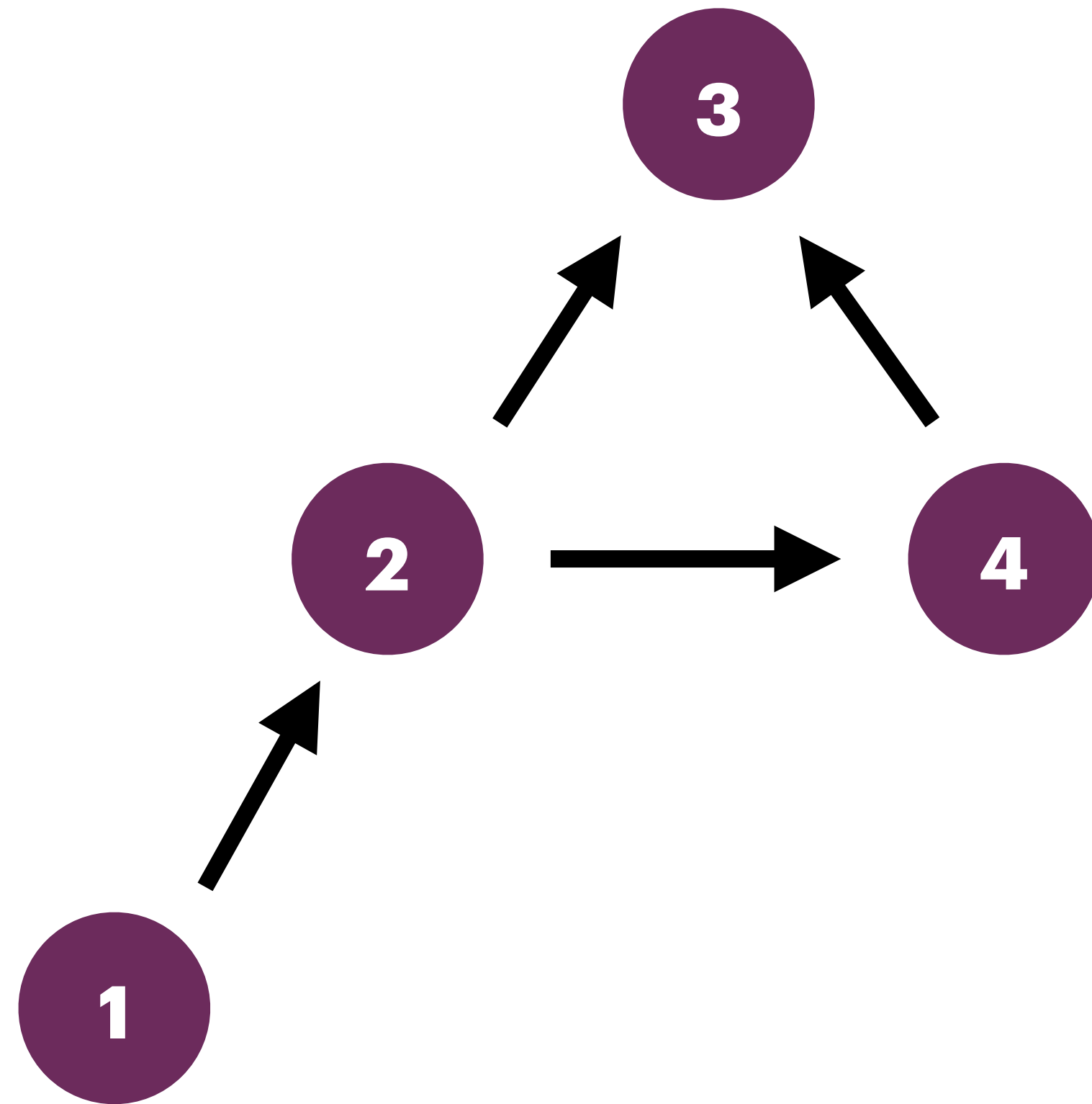
=



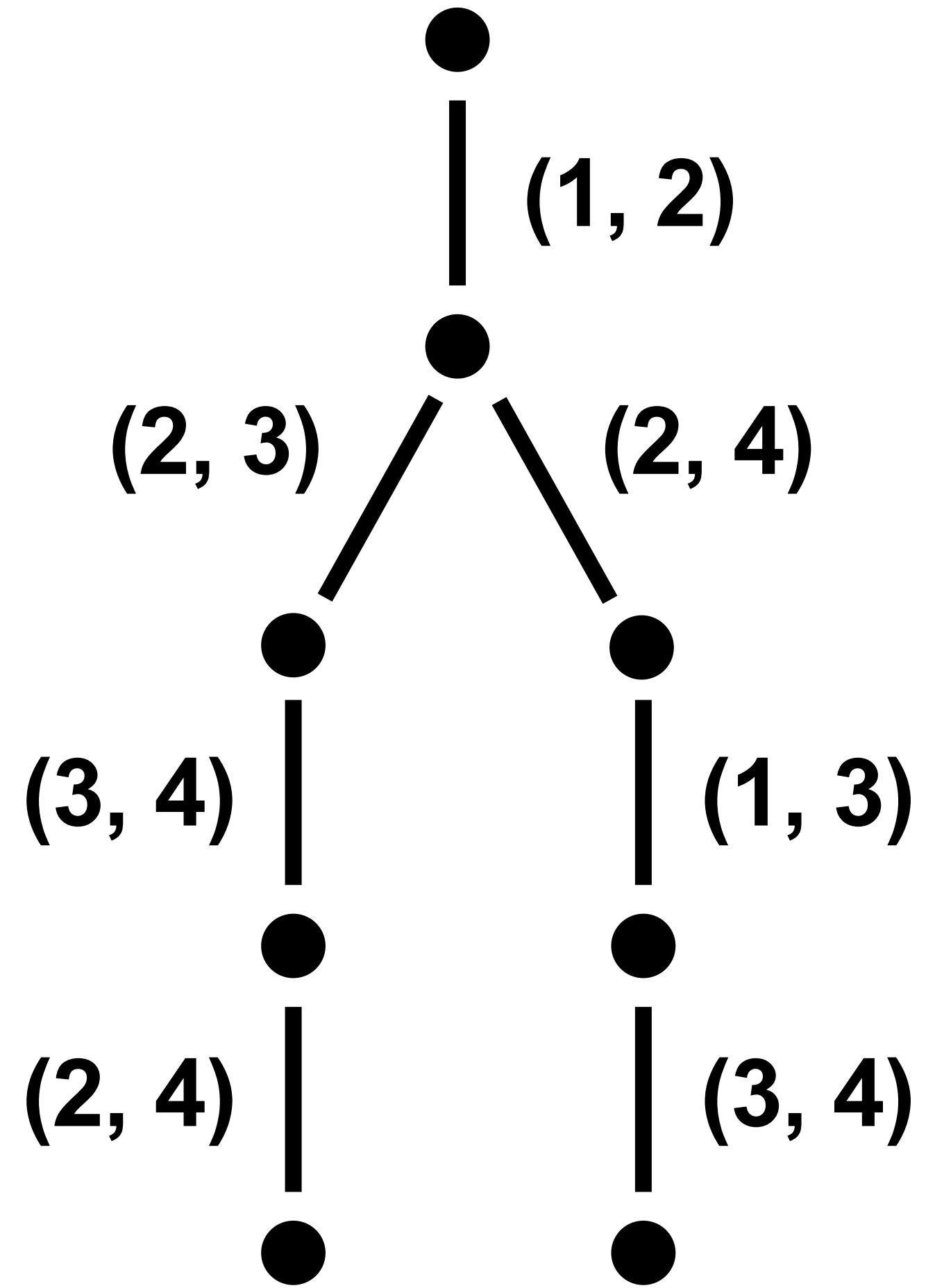
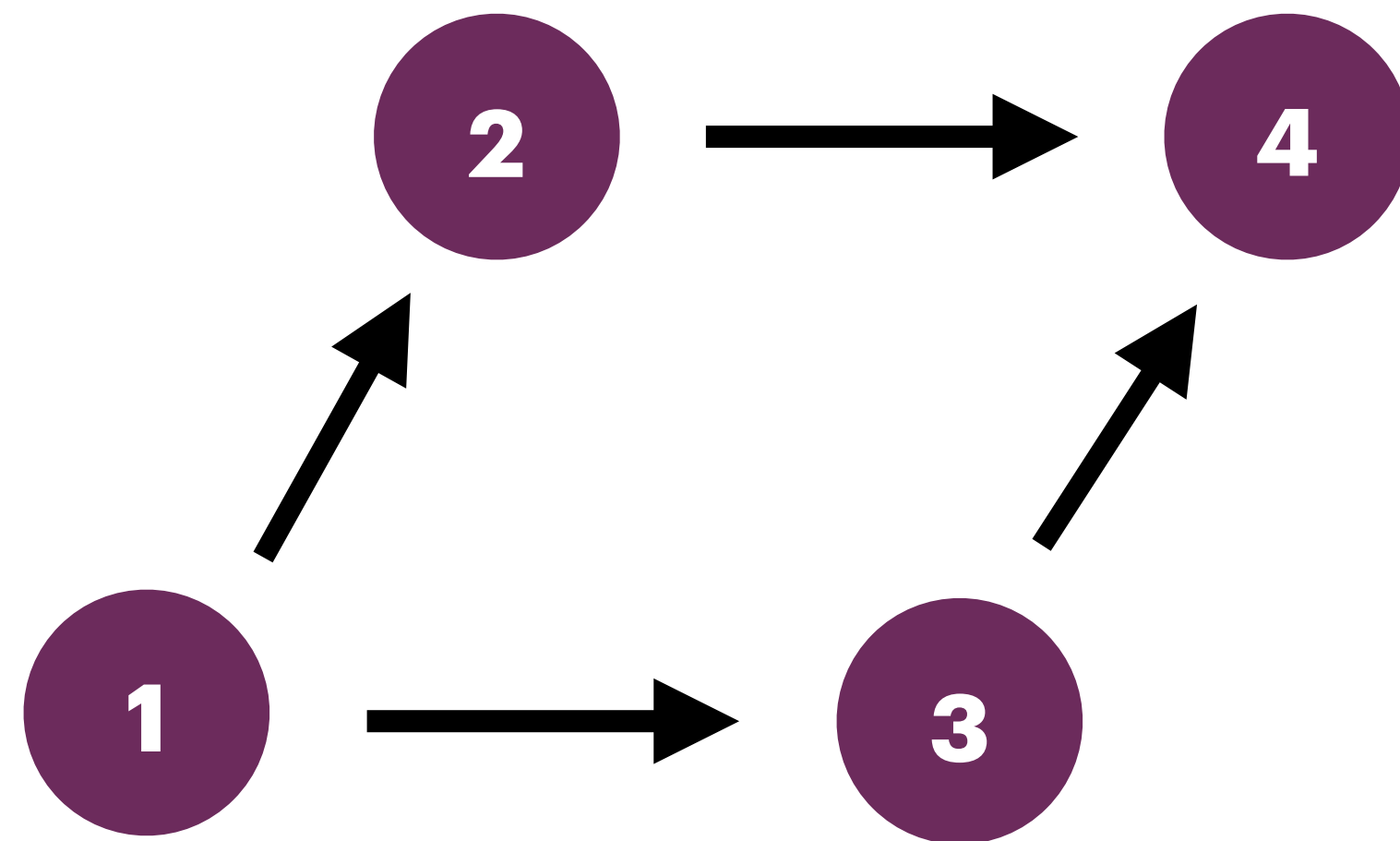
It's all about the vertex labelling



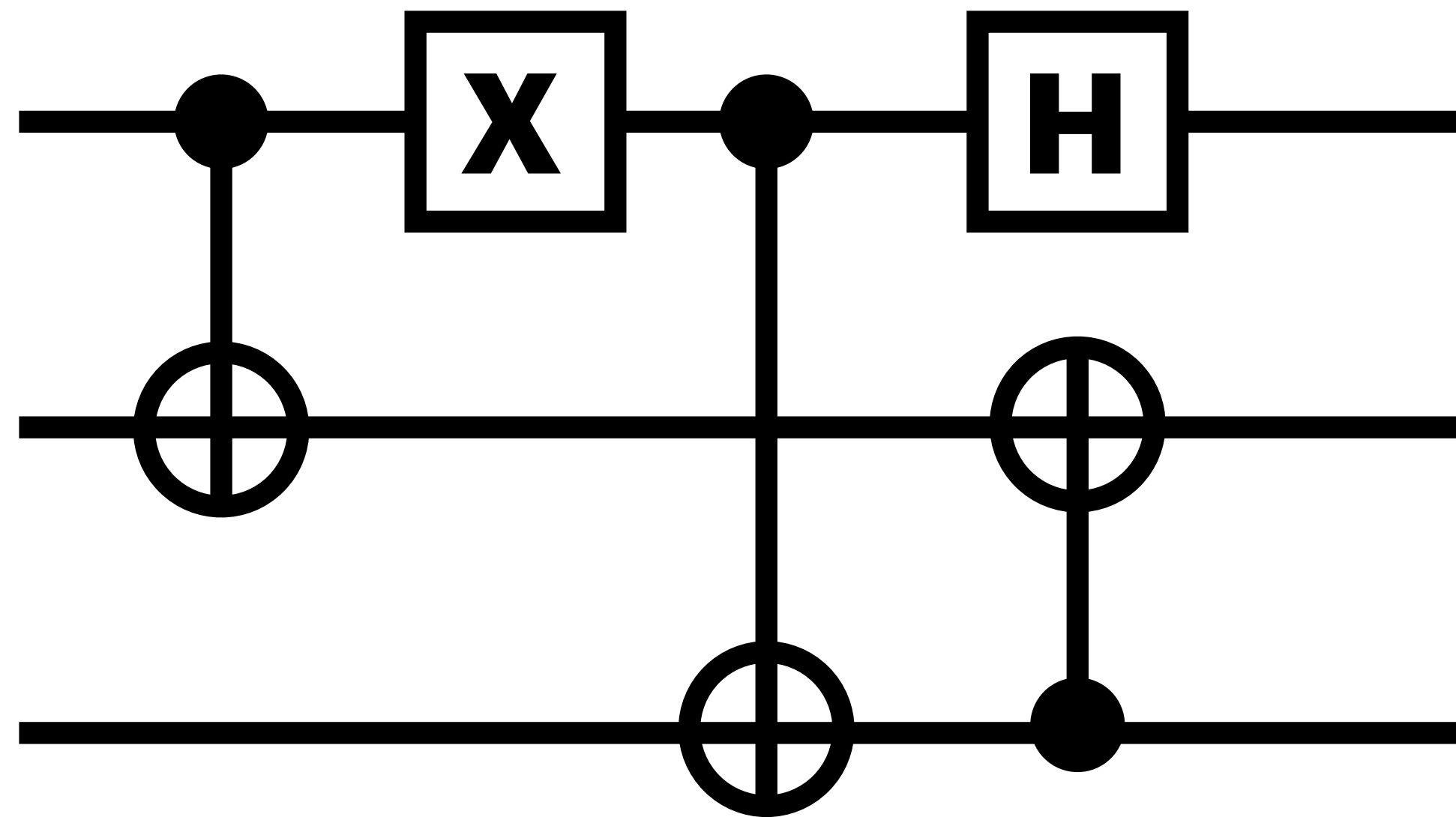
Then edge-by-edge equality



Then edge-by-edge equality

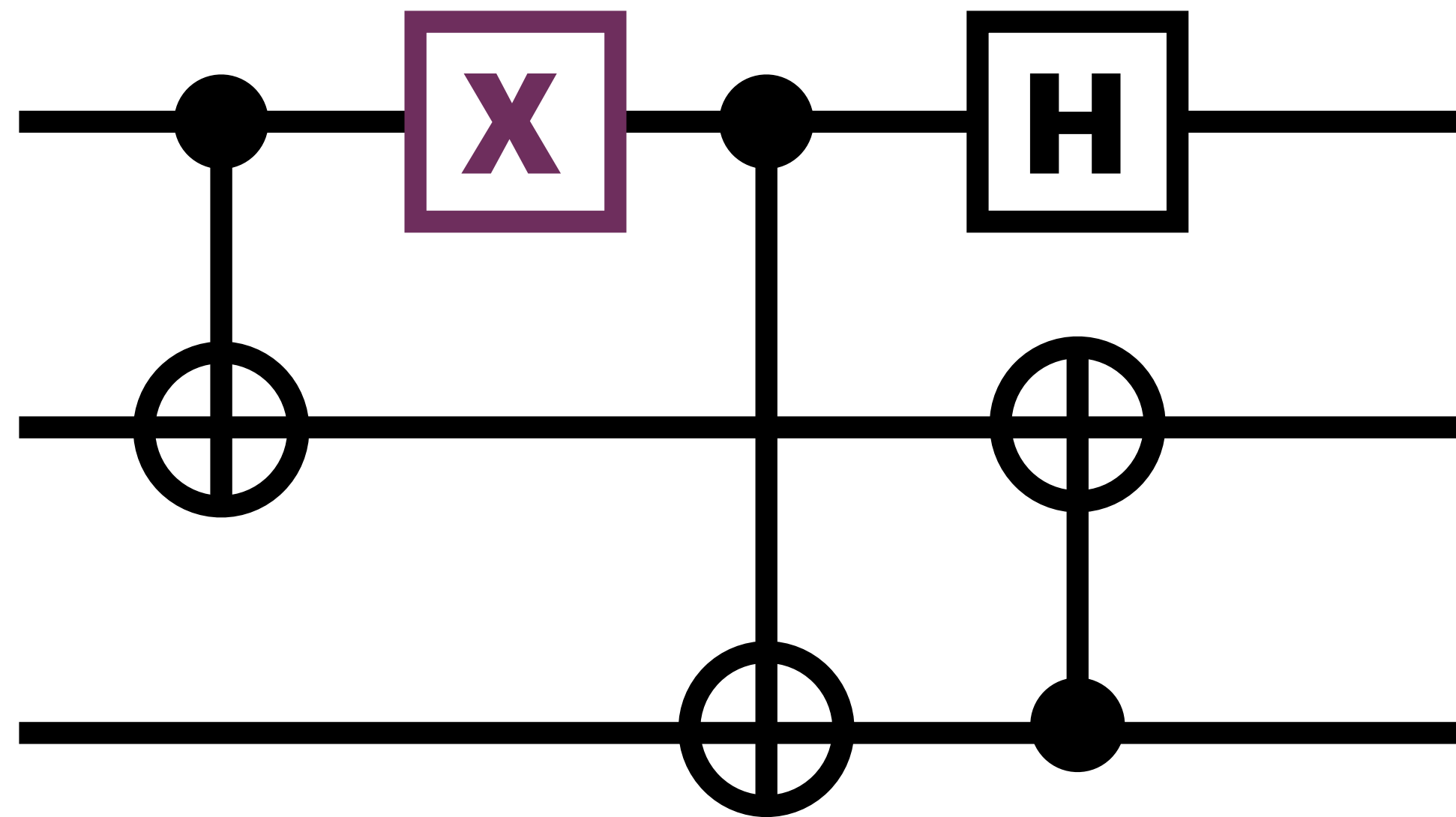


Vertex labelling is easy on circuits!



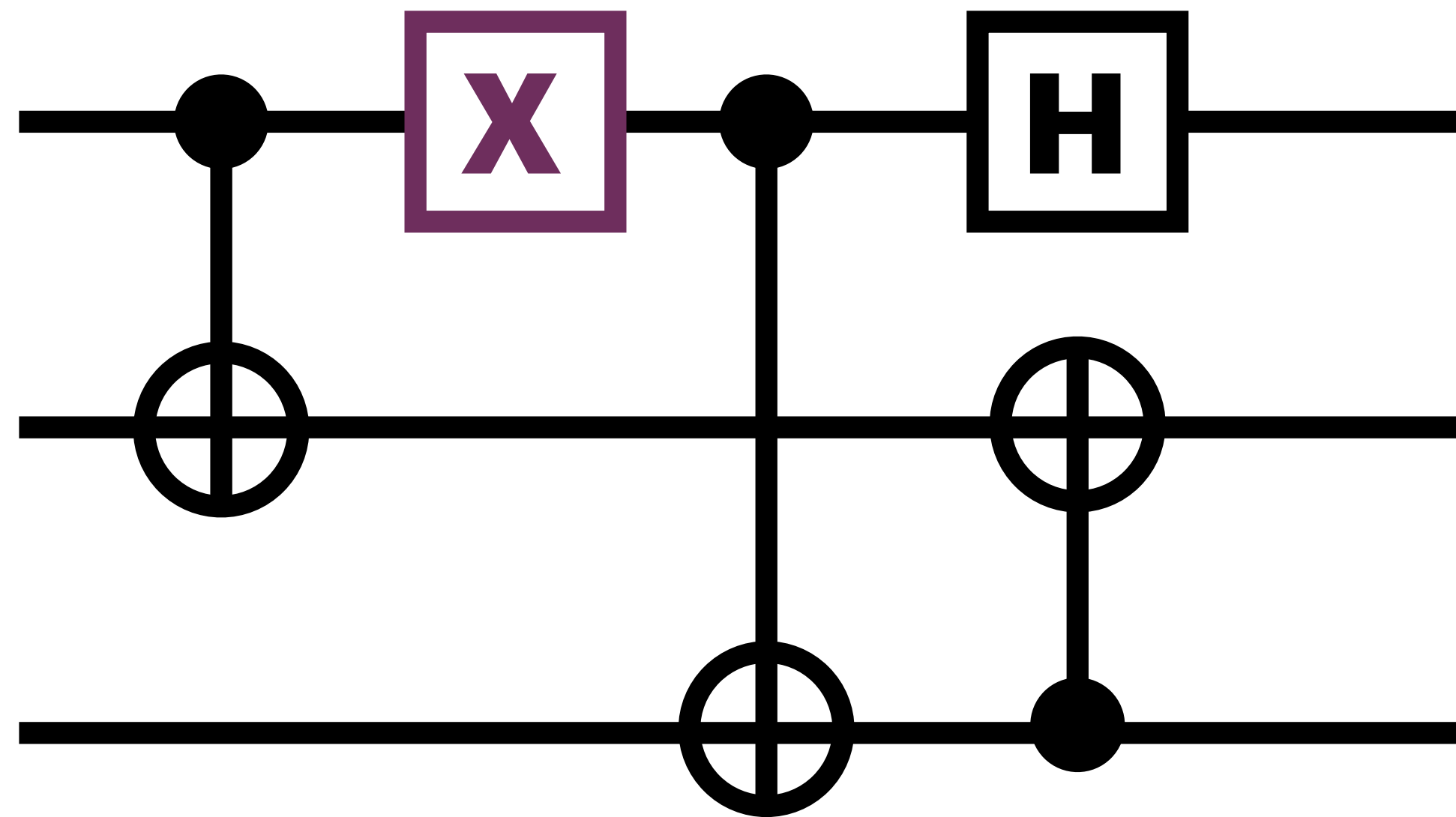
1. Fix root gate
2. Queue of unvisited qubits
3. Traverse every qubit

Vertex labelling is easy on circuits!



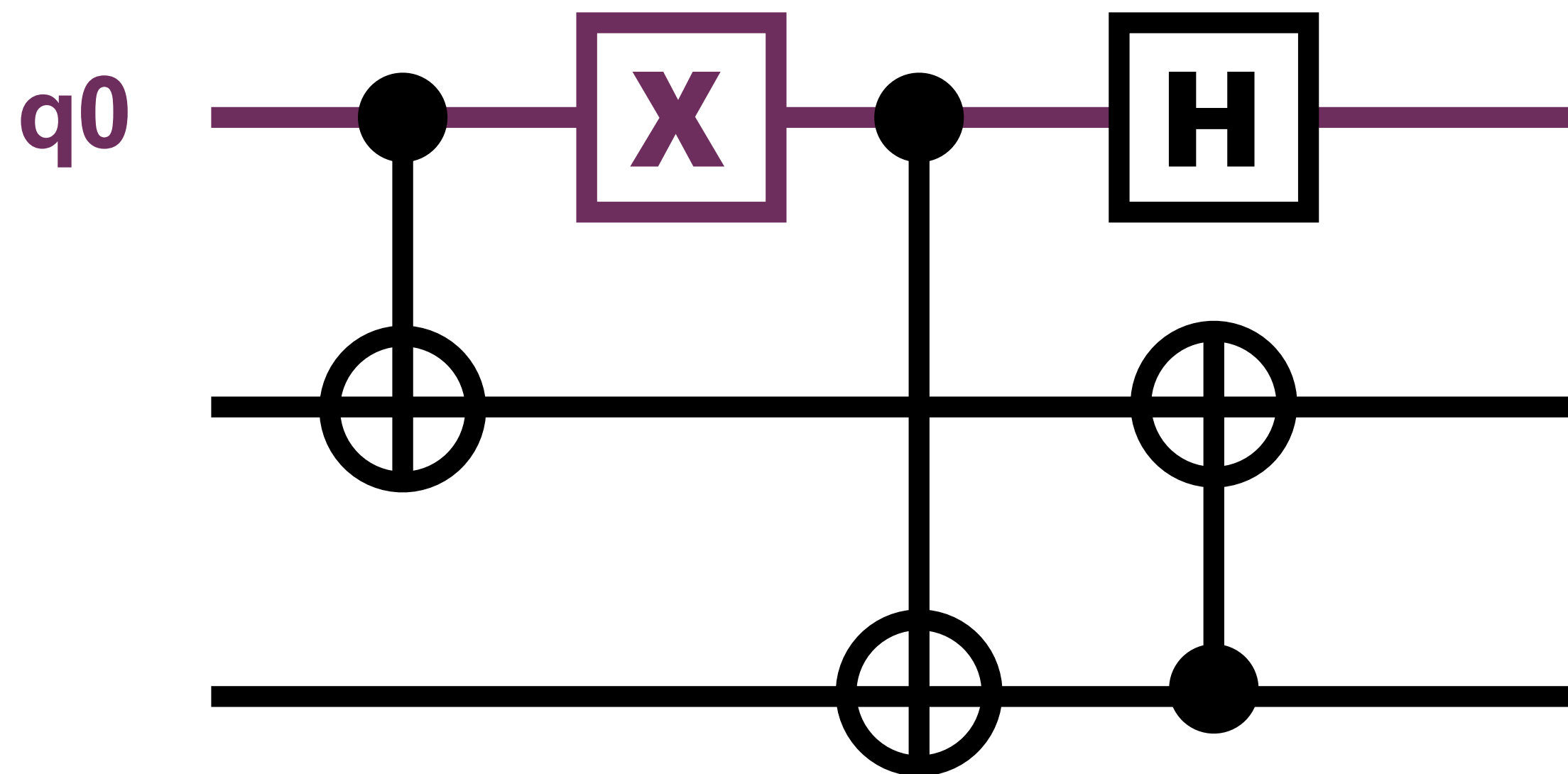
1. **Fix root gate**
2. Queue of unvisited qubits
3. Traverse every qubit

Vertex labelling is easy on circuits!



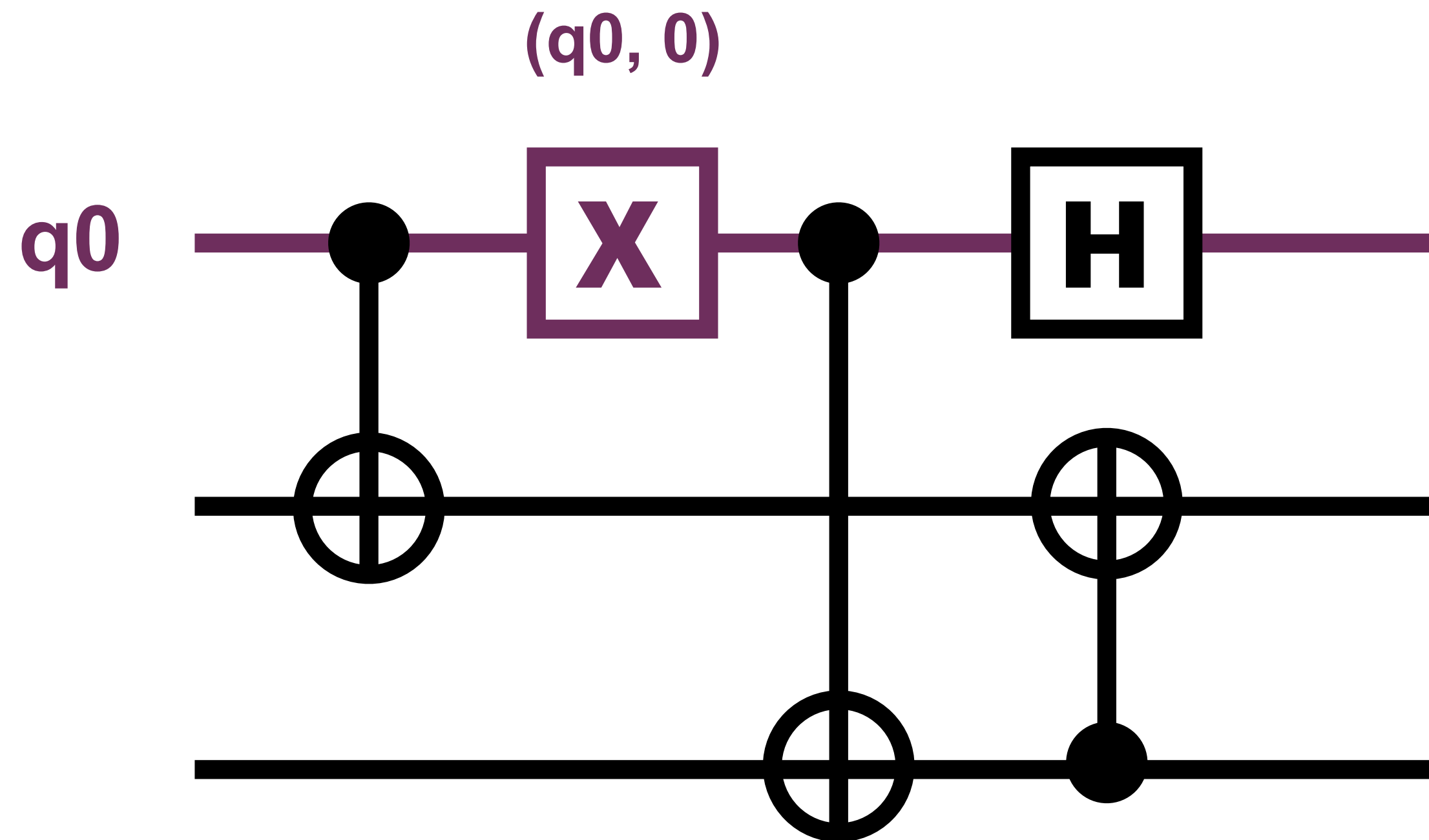
1. Fix root gate
2. **Queue of unvisited qubits**
[q0]
3. Traverse every qubit

Vertex labelling is easy on circuits!



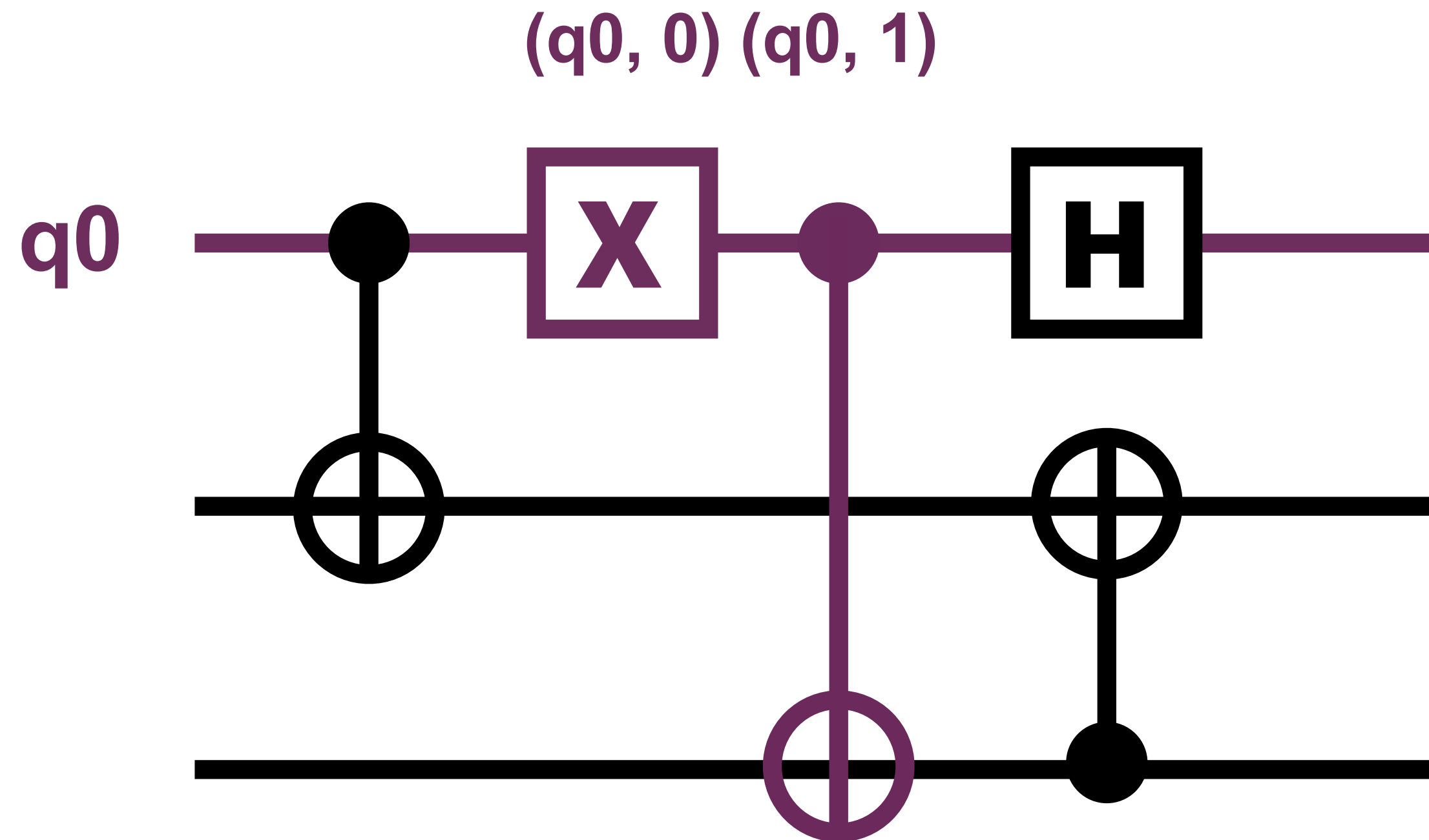
1. Fix root gate
2. Queue of unvisited qubits
[q_0]
3. **Traverse every qubit**

Vertex labelling is easy on circuits!



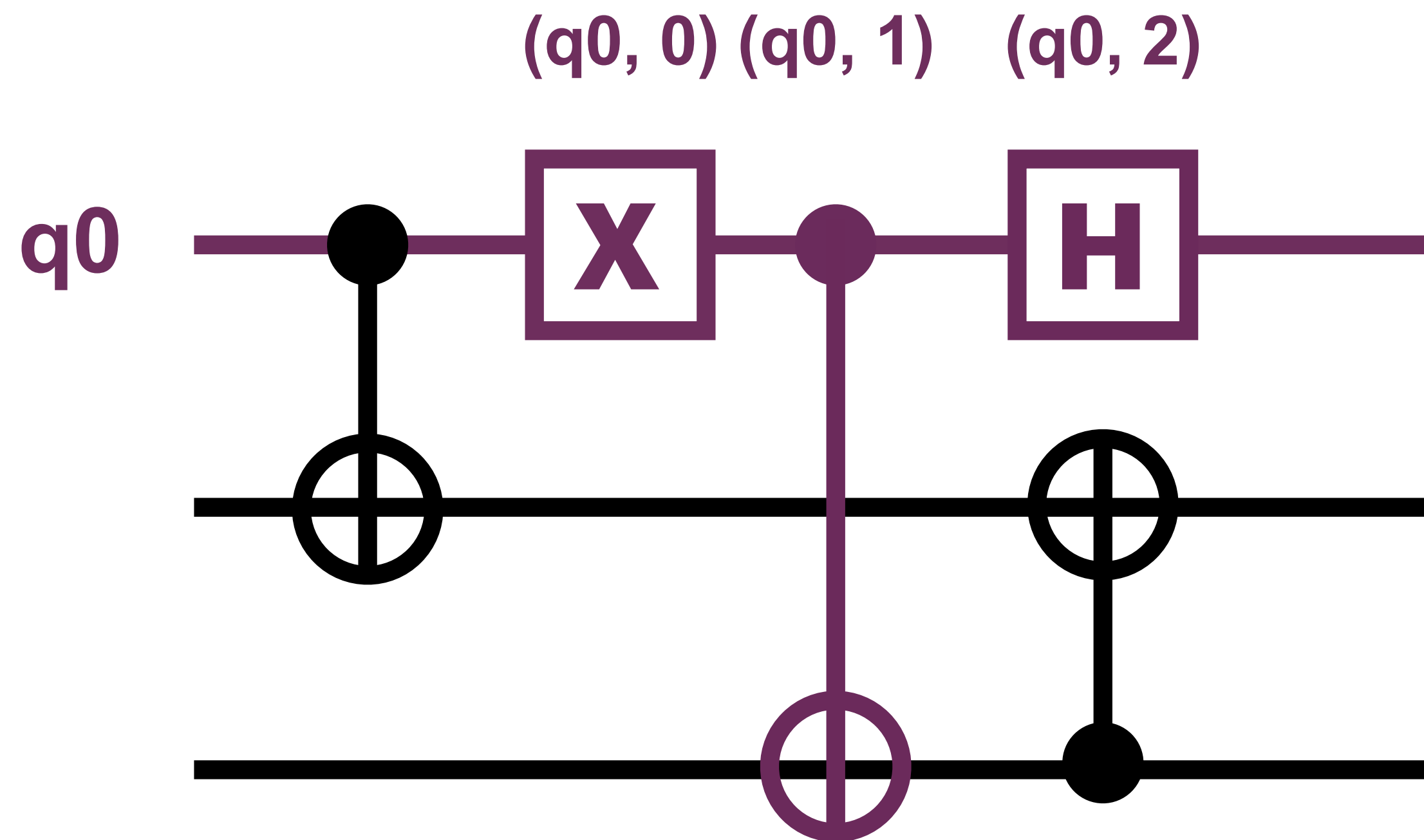
1. Fix root gate
2. Queue of unvisited qubits
[q0]
3. **Traverse every qubit**

Vertex labelling is easy on circuits!



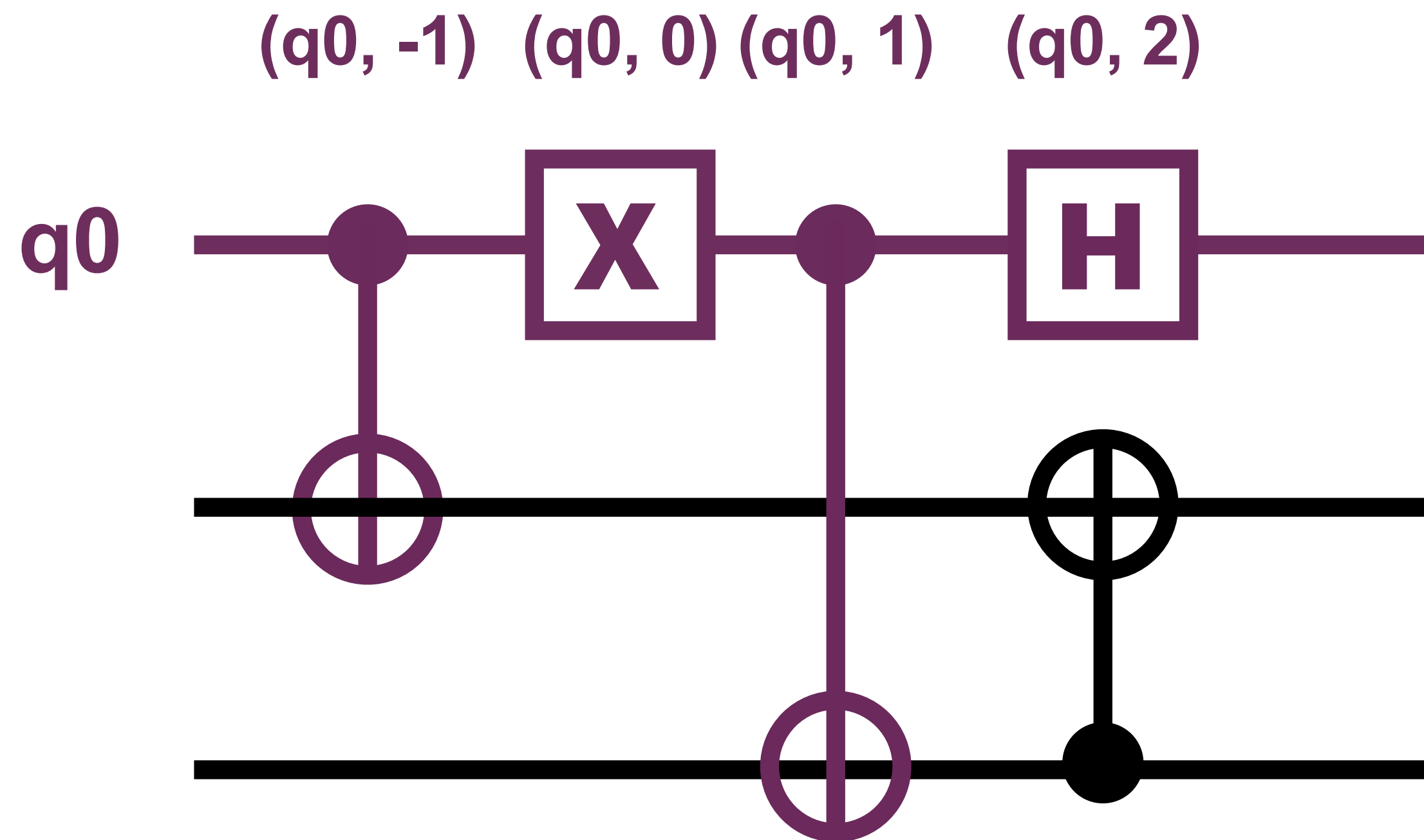
1. Fix root gate
2. Queue of unvisited qubits
[q_0, q_1]
3. **Traverse every qubit**

Vertex labelling is easy on circuits!



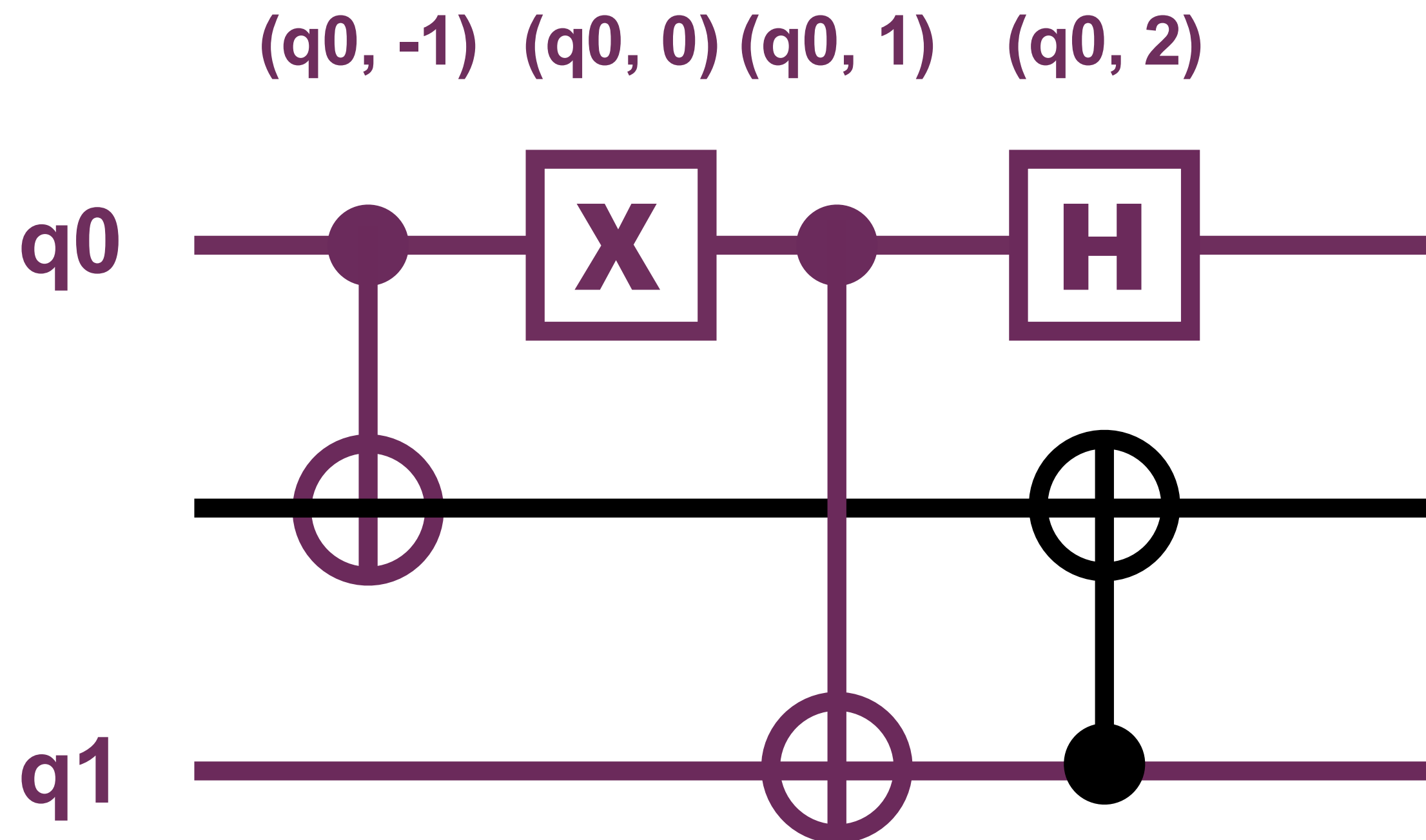
1. Fix root gate
2. Queue of unvisited qubits
[q_0, q_1]
3. **Traverse every qubit**

Vertex labelling is easy on circuits!



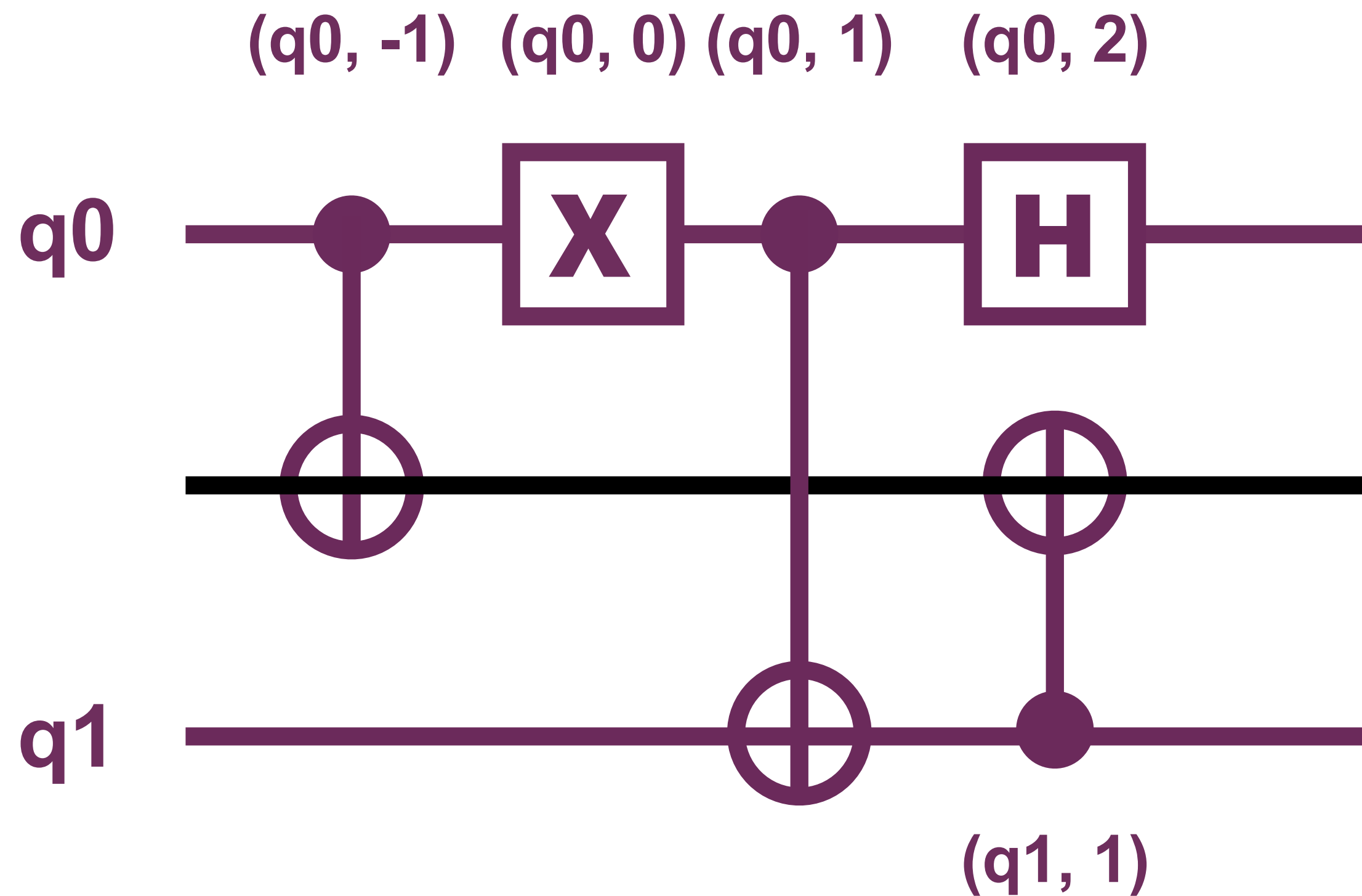
1. Fix root gate
2. Queue of unvisited qubits
 $[q_0, q_1, q_2]$
3. **Traverse every qubit**

Vertex labelling is easy on circuits!



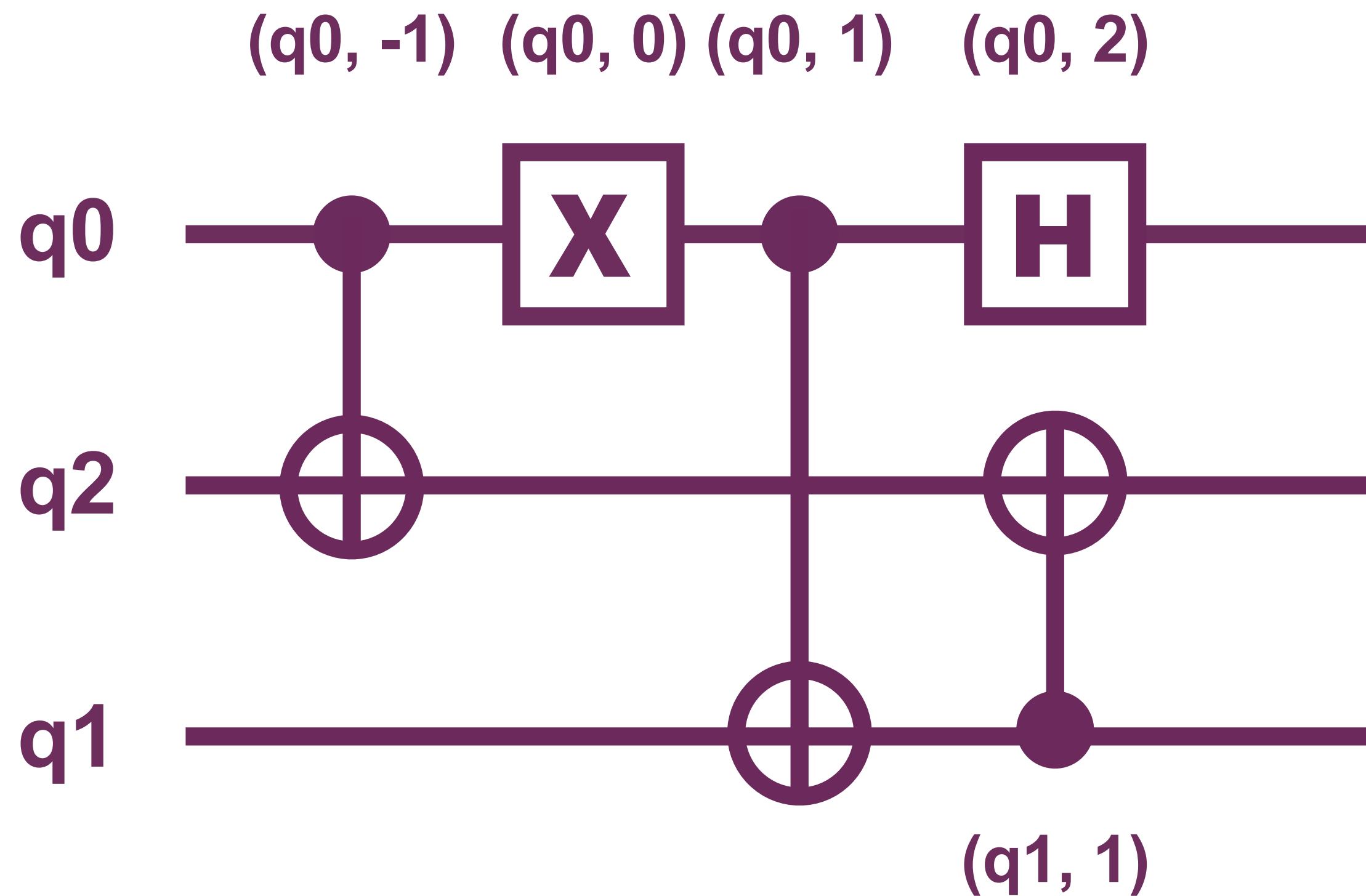
1. Fix root gate
2. **Queue of unvisited qubits**
[q_0, q_1, q_2]
3. Traverse every qubit

Vertex labelling is easy on circuits!



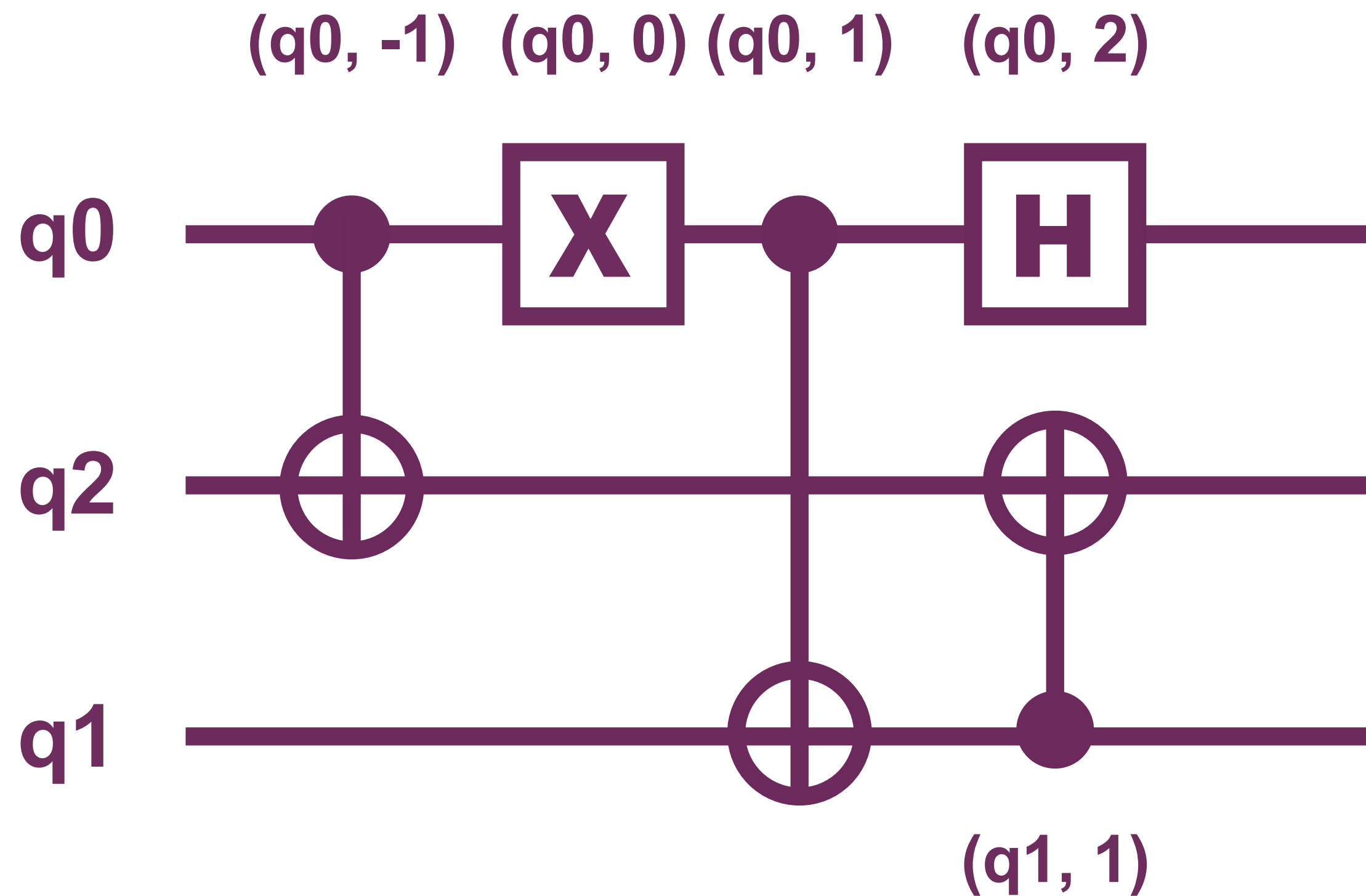
1. Fix root gate
2. Queue of unvisited qubits
[q_0, q_1, q_2]
3. **Traverse every qubit**

Vertex labelling is easy on circuits!



1. Fix root gate
2. **Queue of unvisited qubits**
[q_0, q_1, q_2]
3. Traverse every qubit

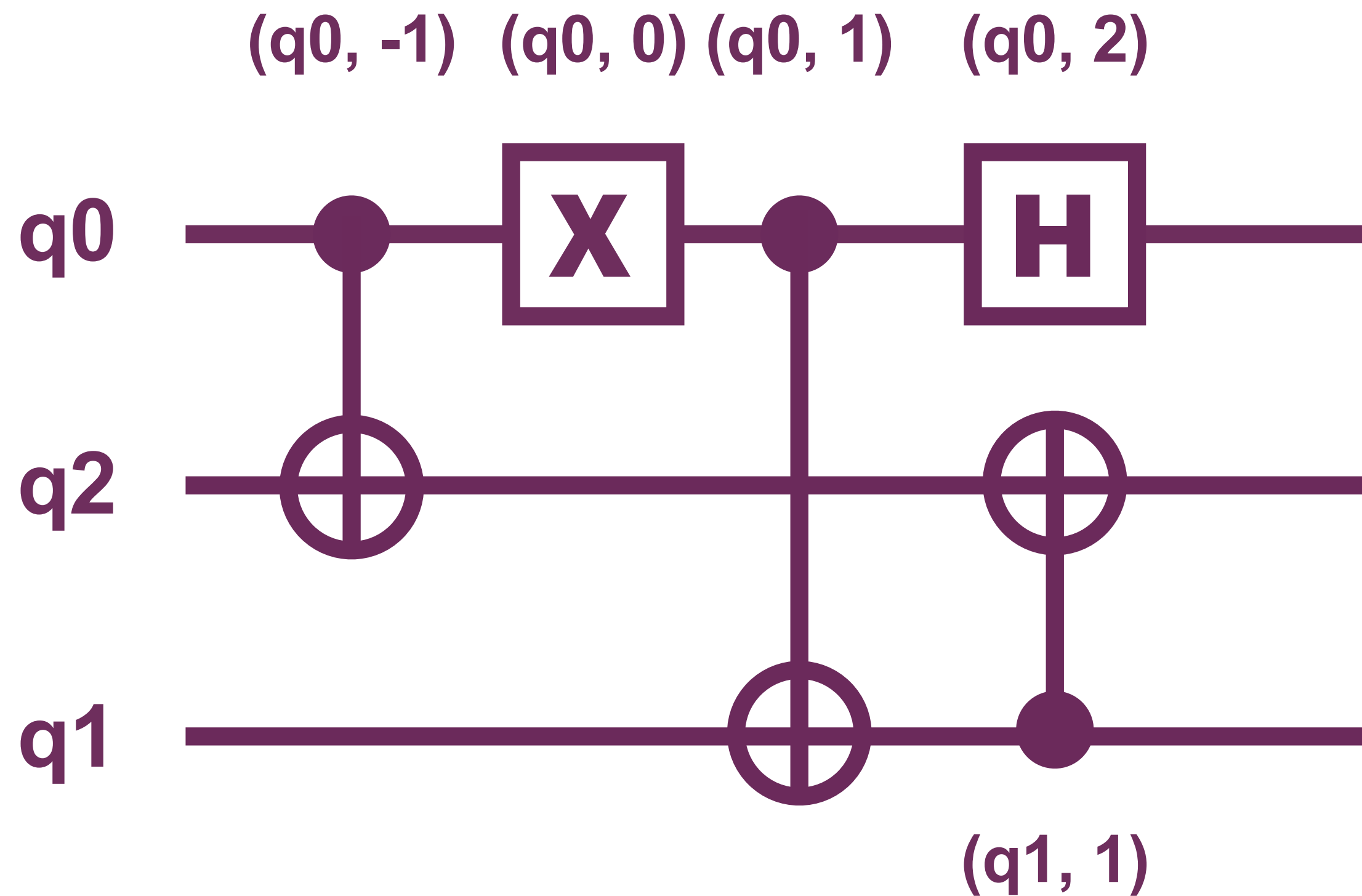
Vertex labelling is easy on circuits!



► For fixed root, labelling is unique

1. Fix root gate
2. Queue of unvisited qubits
[q_0, q_1, q_2]
3. Traverse every qubit

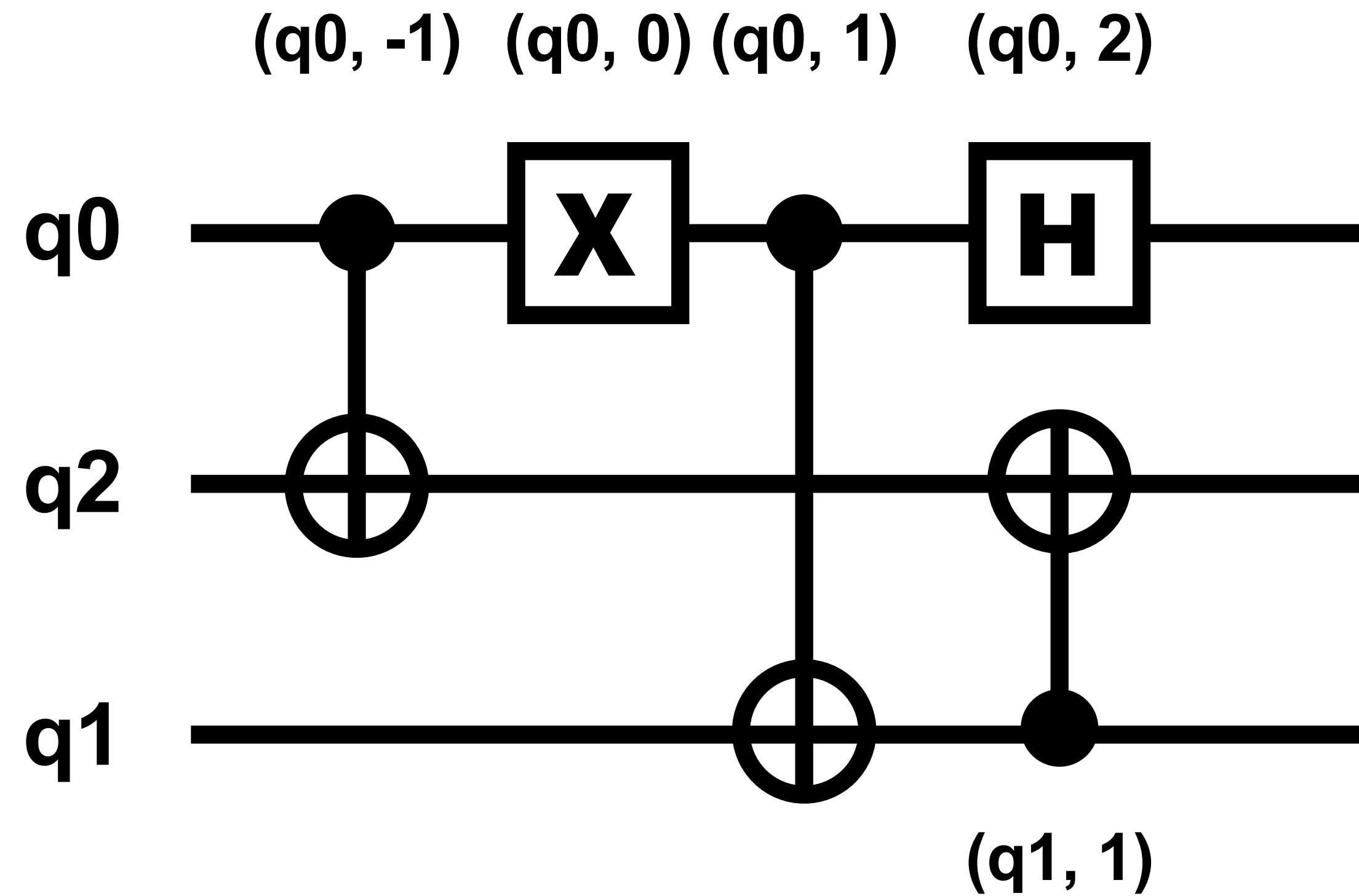
Vertex labelling is easy on circuits!



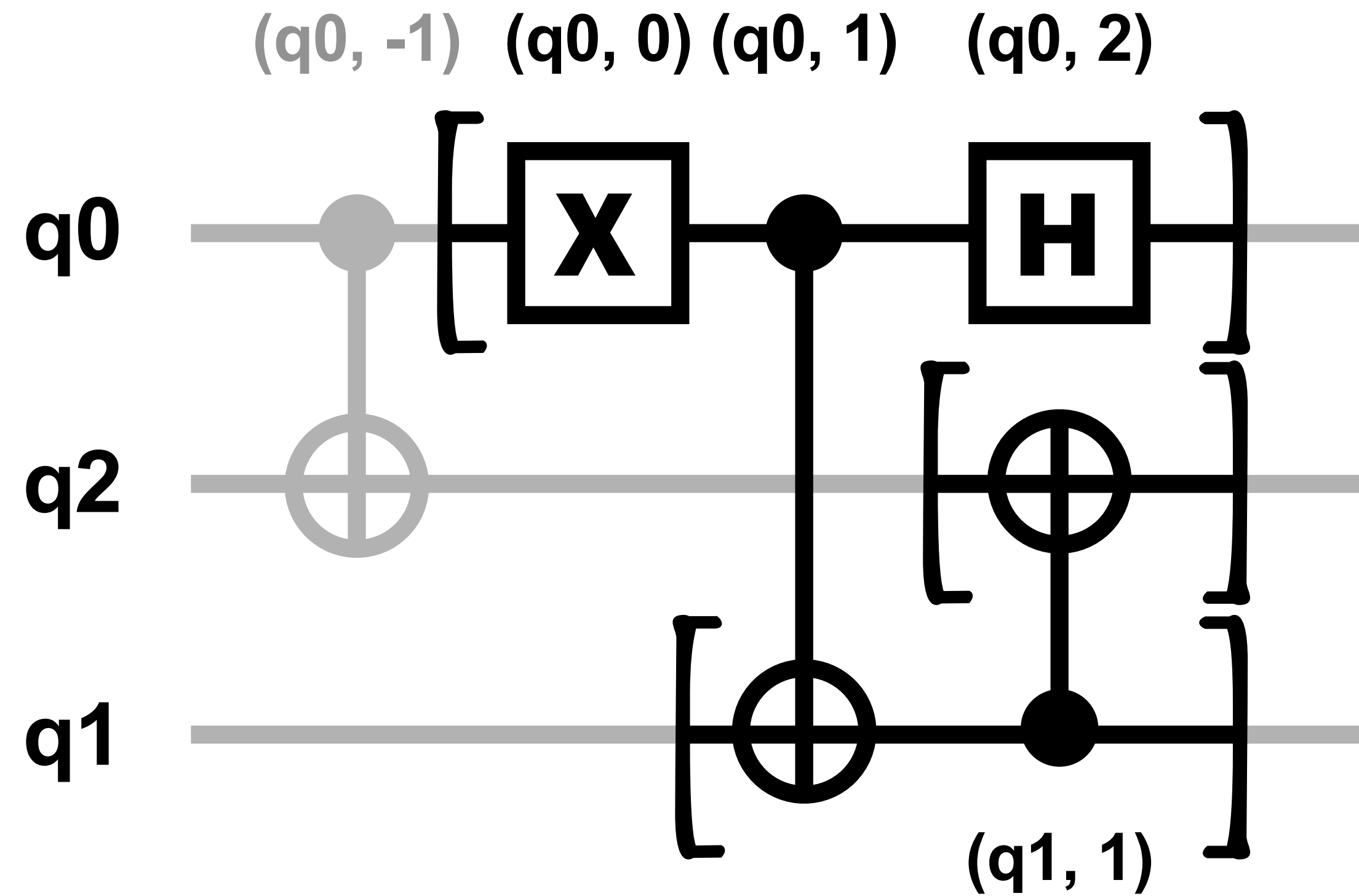
- ▶ For fixed root, labelling is unique
- ▶ n labelling in total

1. Fix root gate
2. Queue of unvisited qubits
[q0, q1, q2]
3. Traverse every qubit

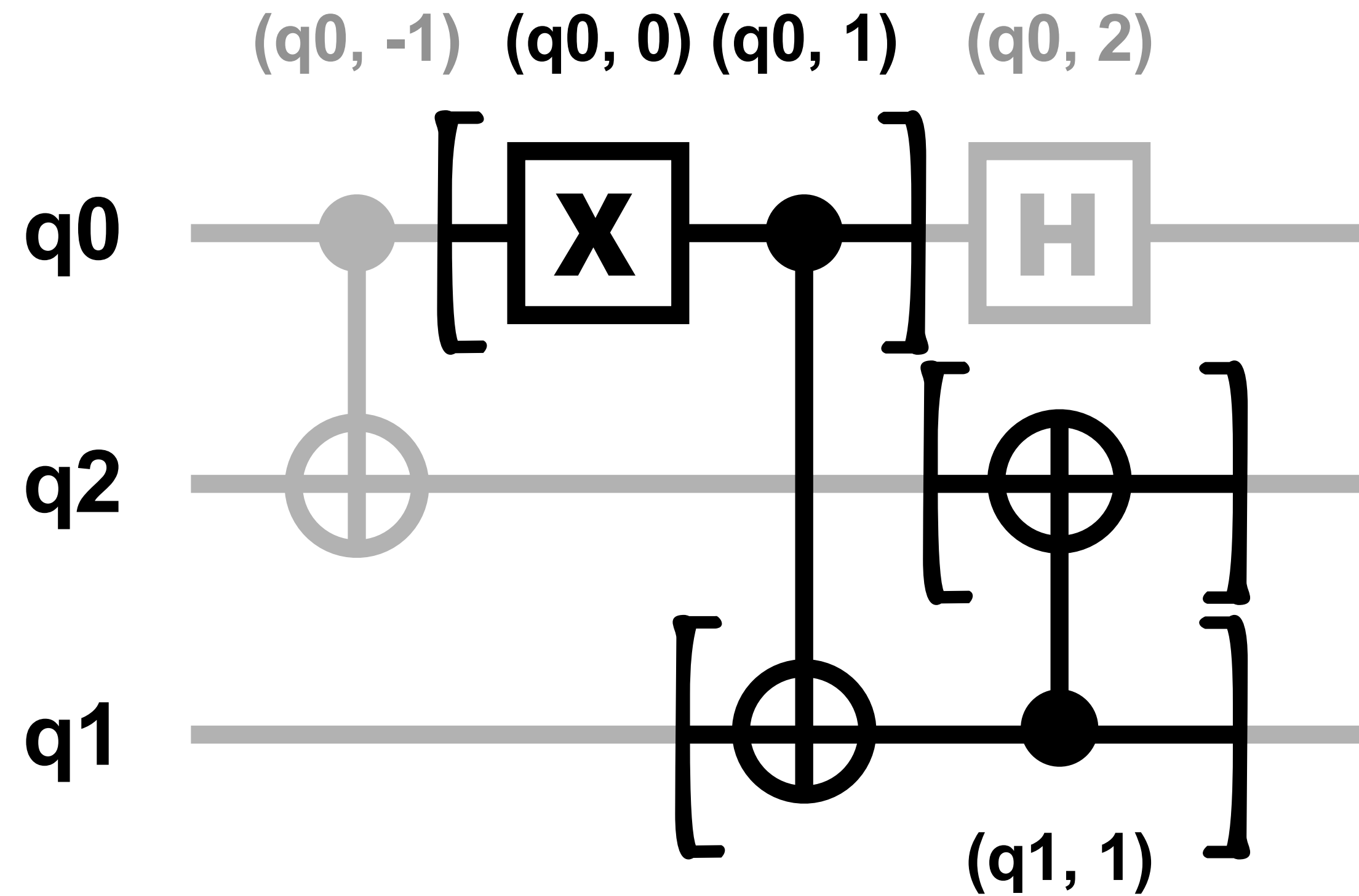
What about labellings of sub-circuits?



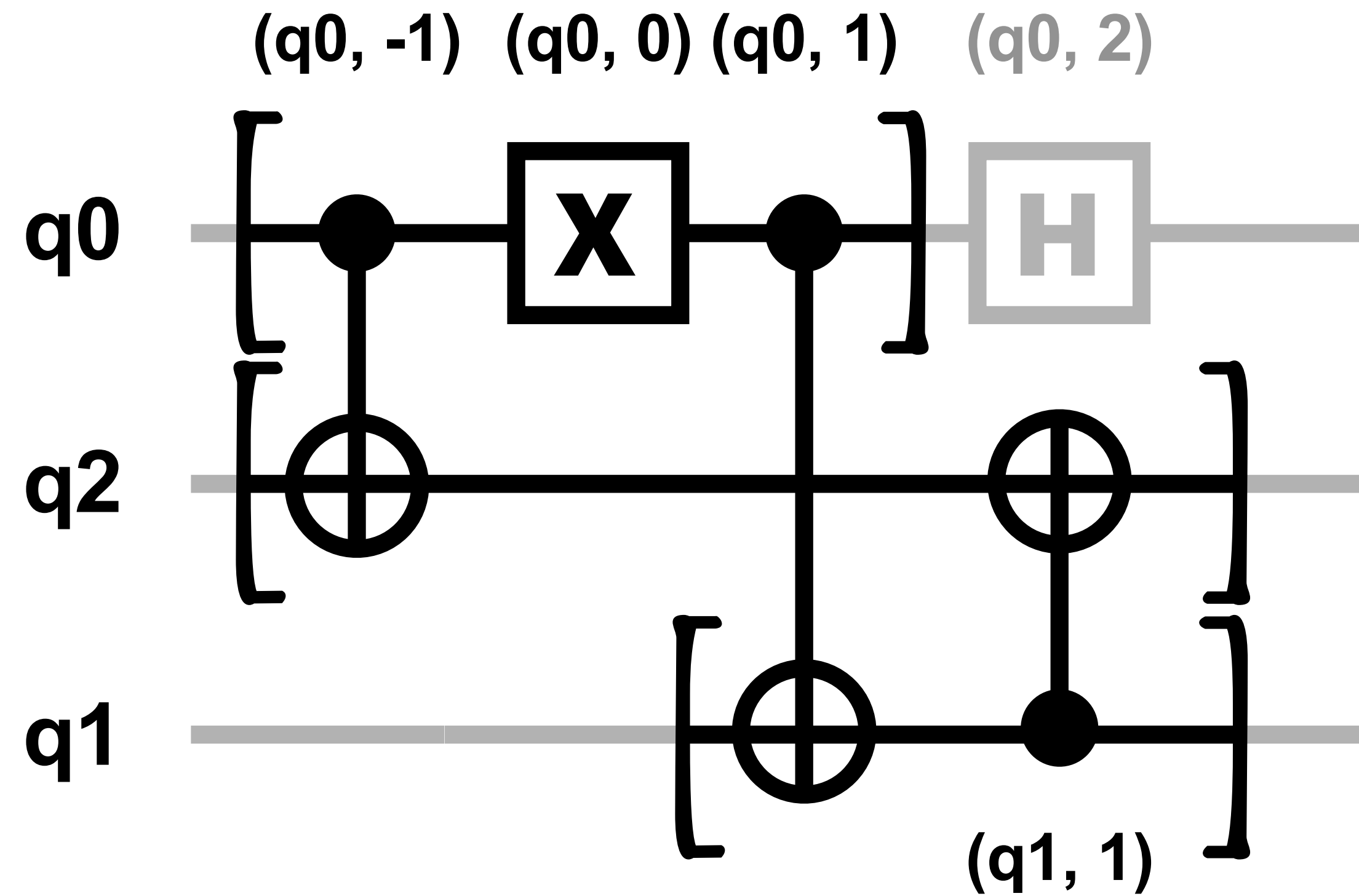
What about labellings of sub-circuits?



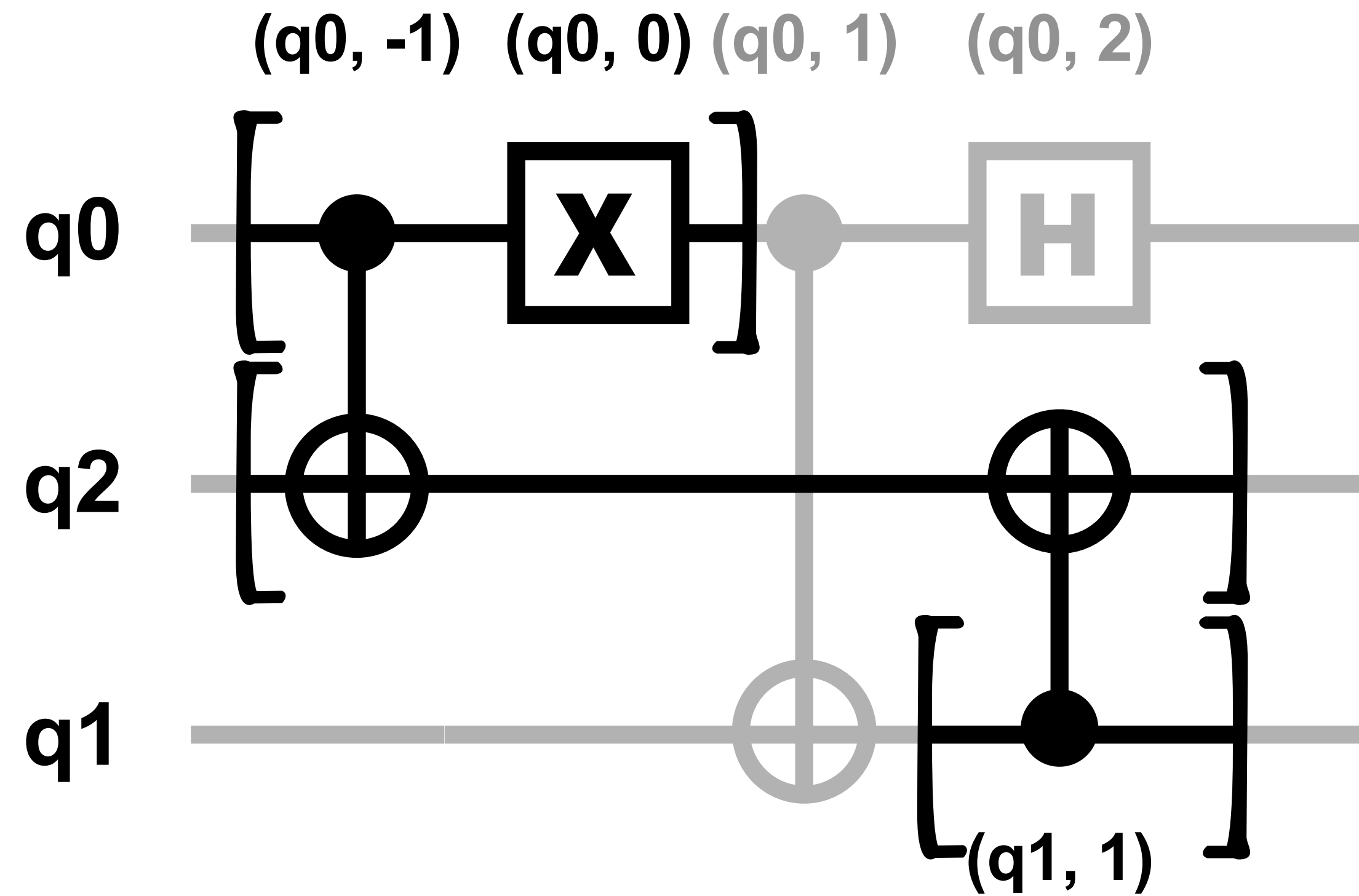
What about labellings of sub-circuits?



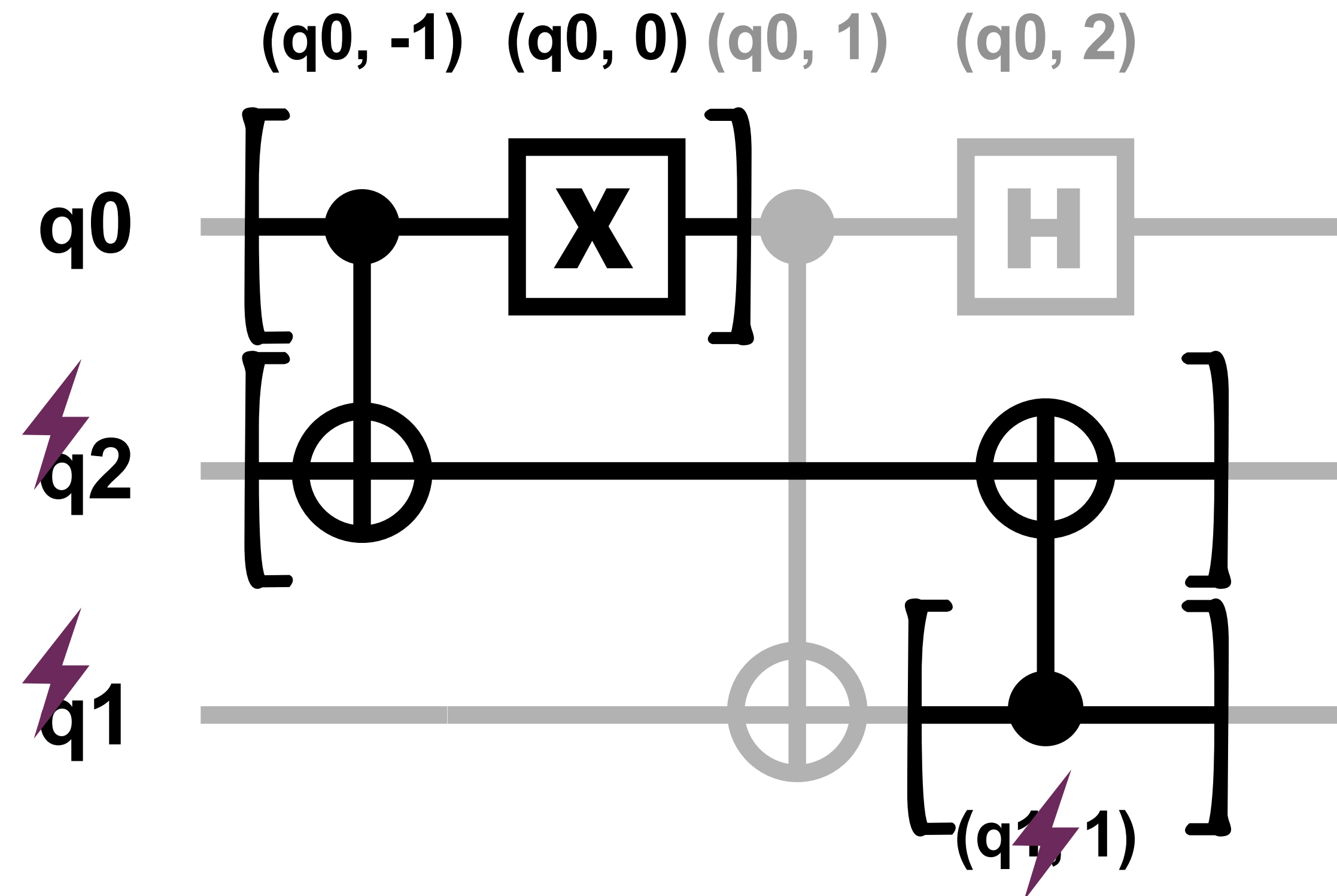
What about labellings of sub-circuits?



What about labellings of sub-circuits?

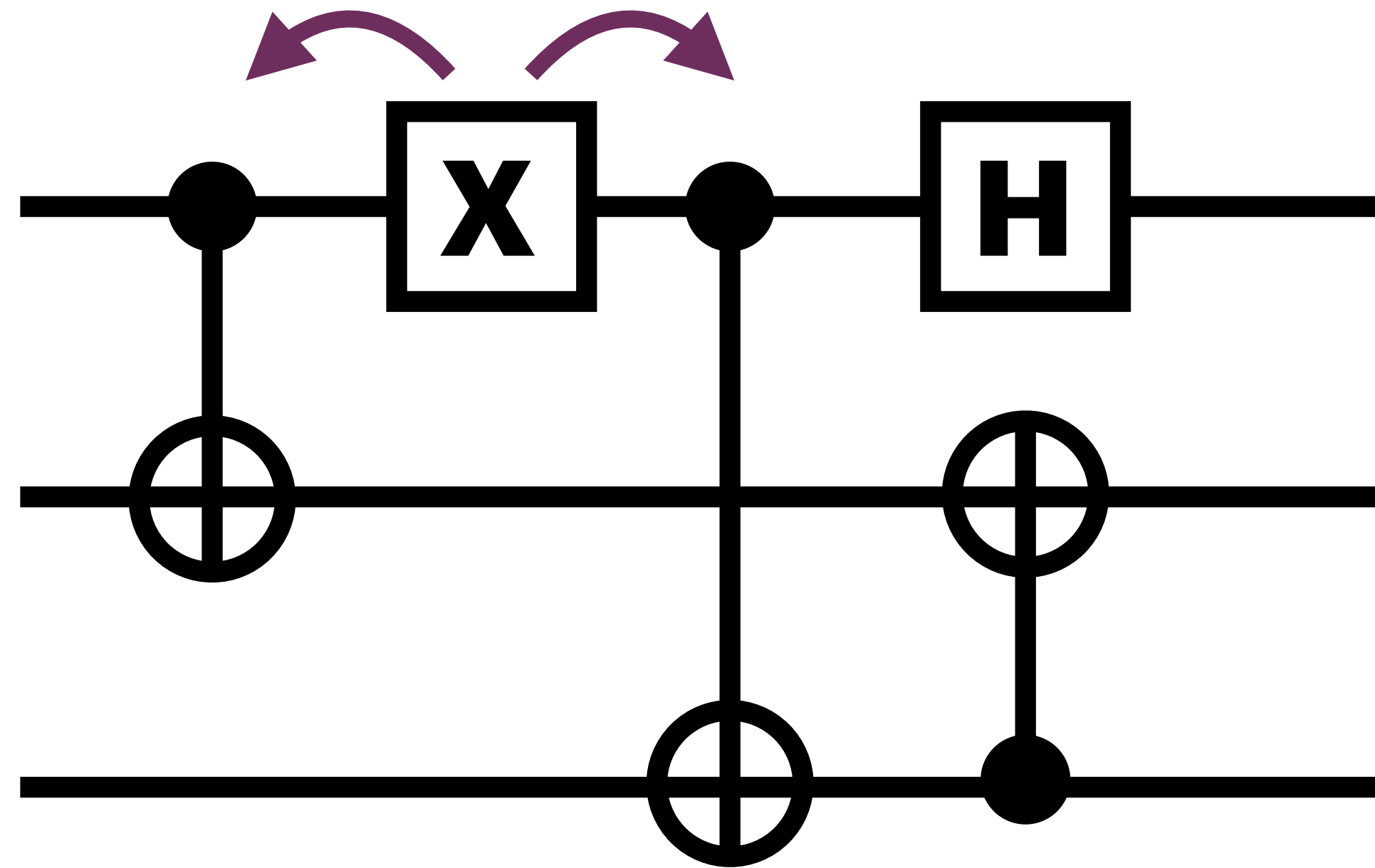


What about labellings of sub-circuits?

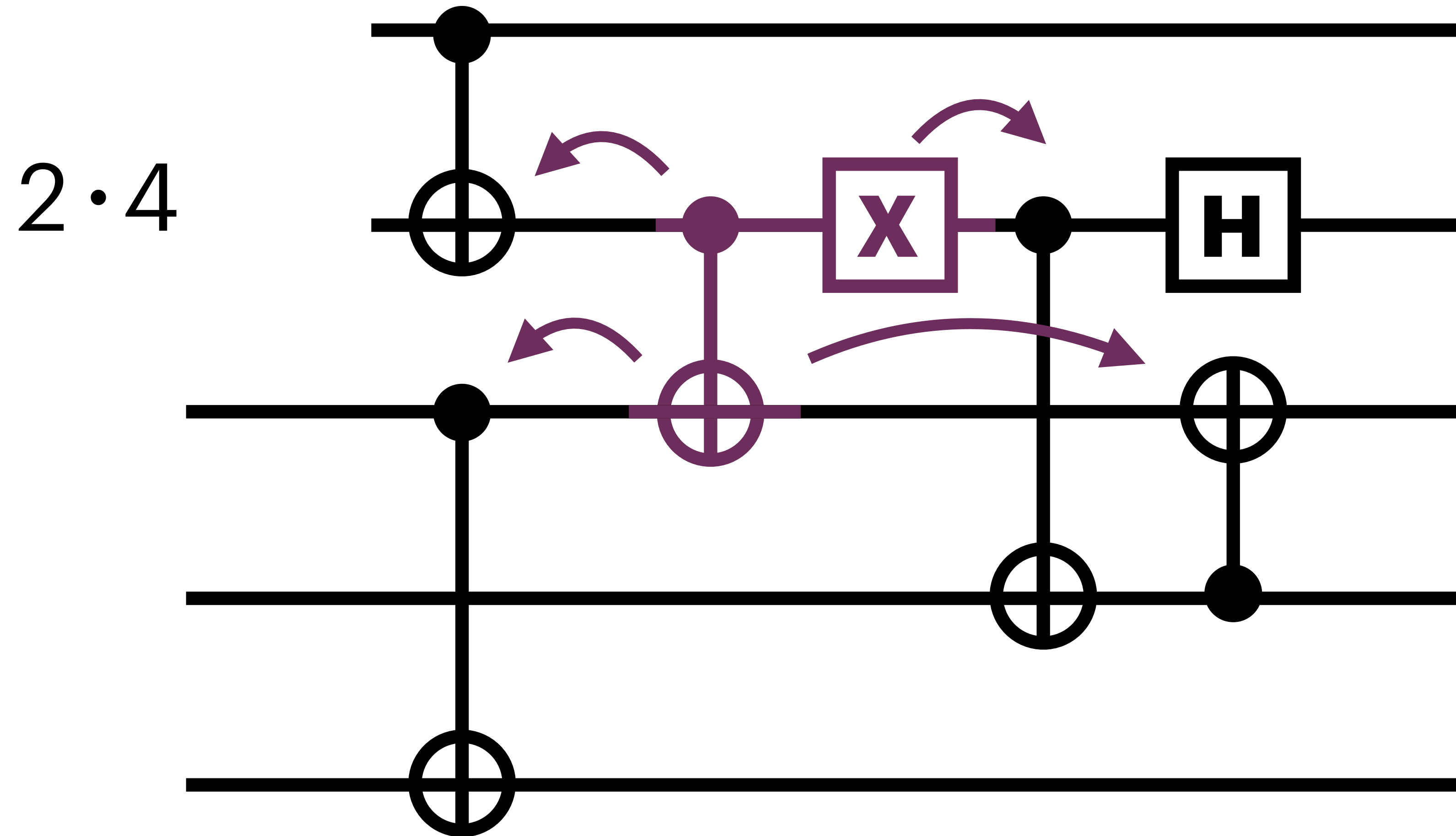


We can enumerate them!

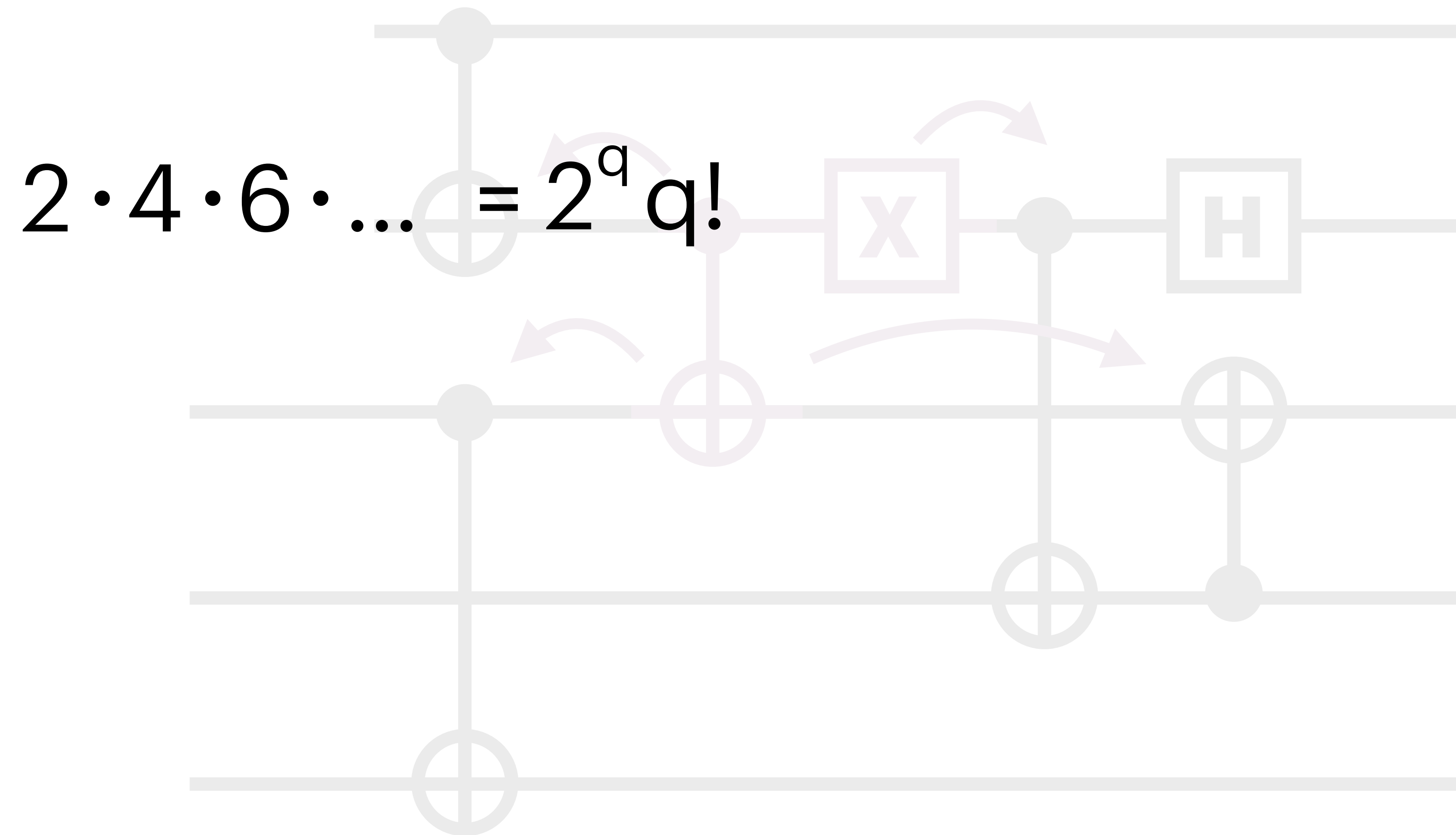
2



We can enumerate them!

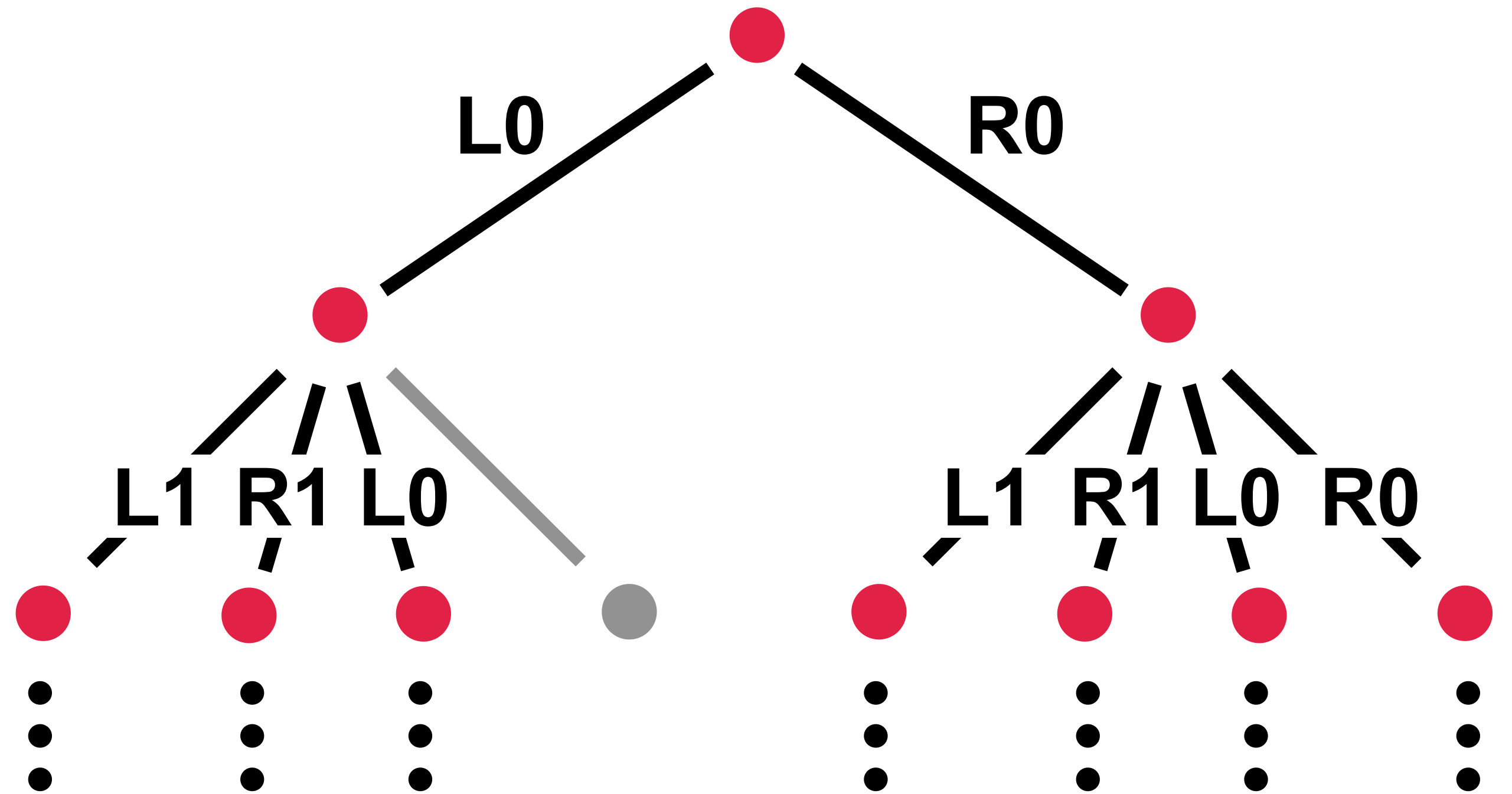


We can enumerate them!

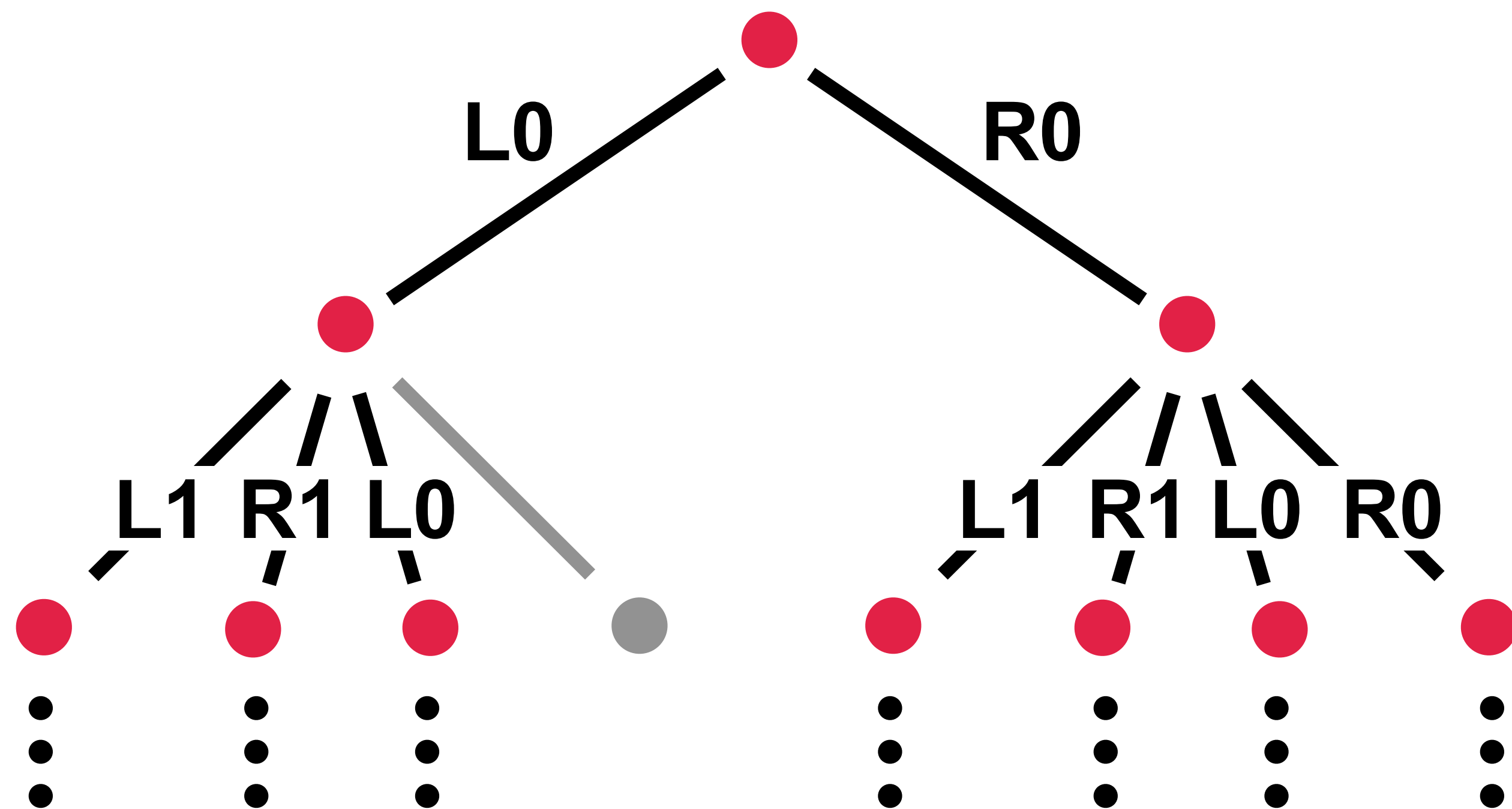
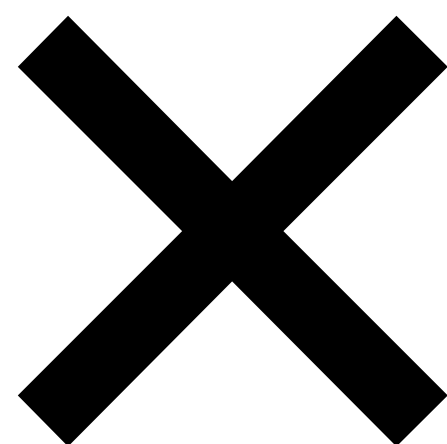
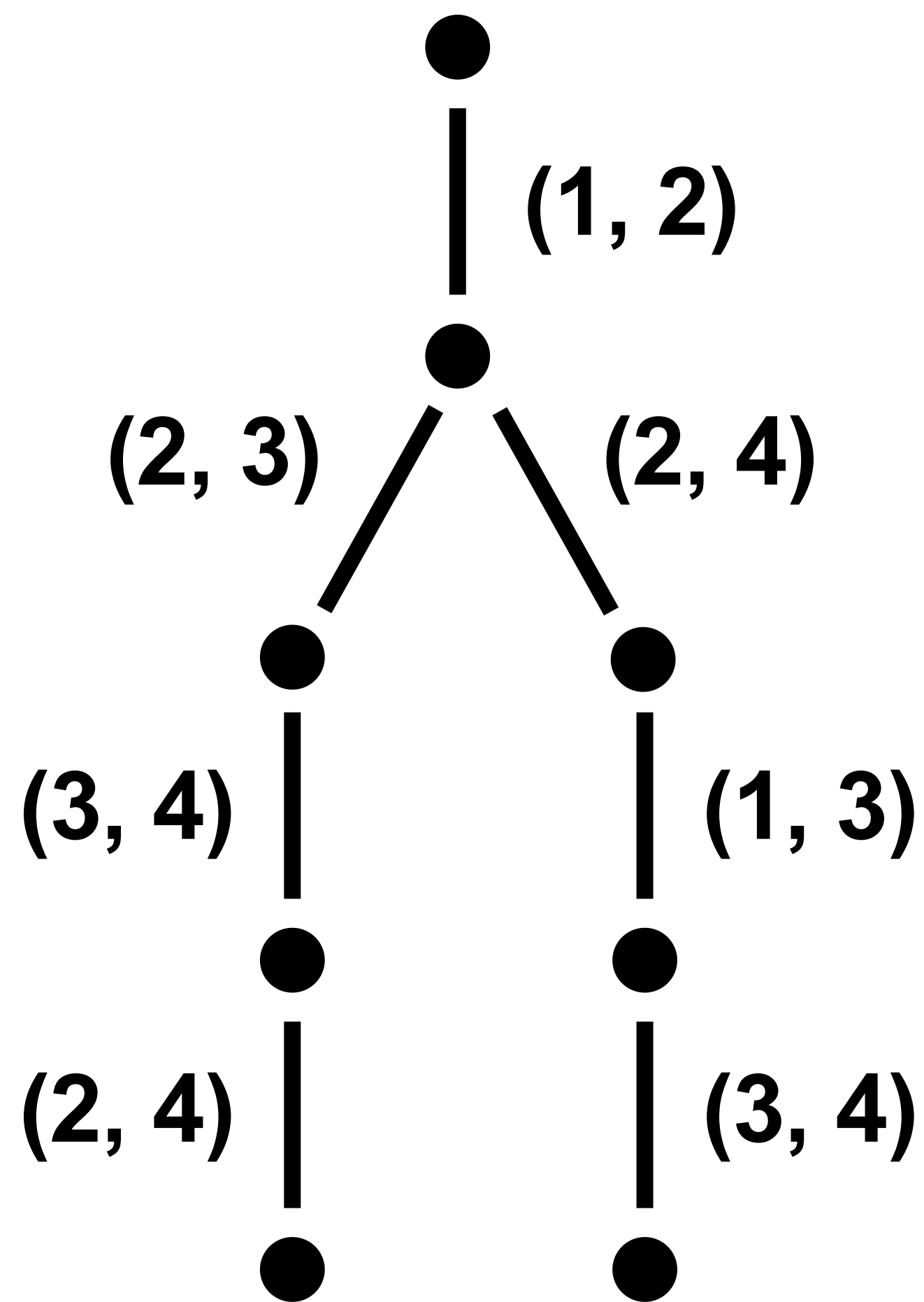


We can enumerate them in a tree!

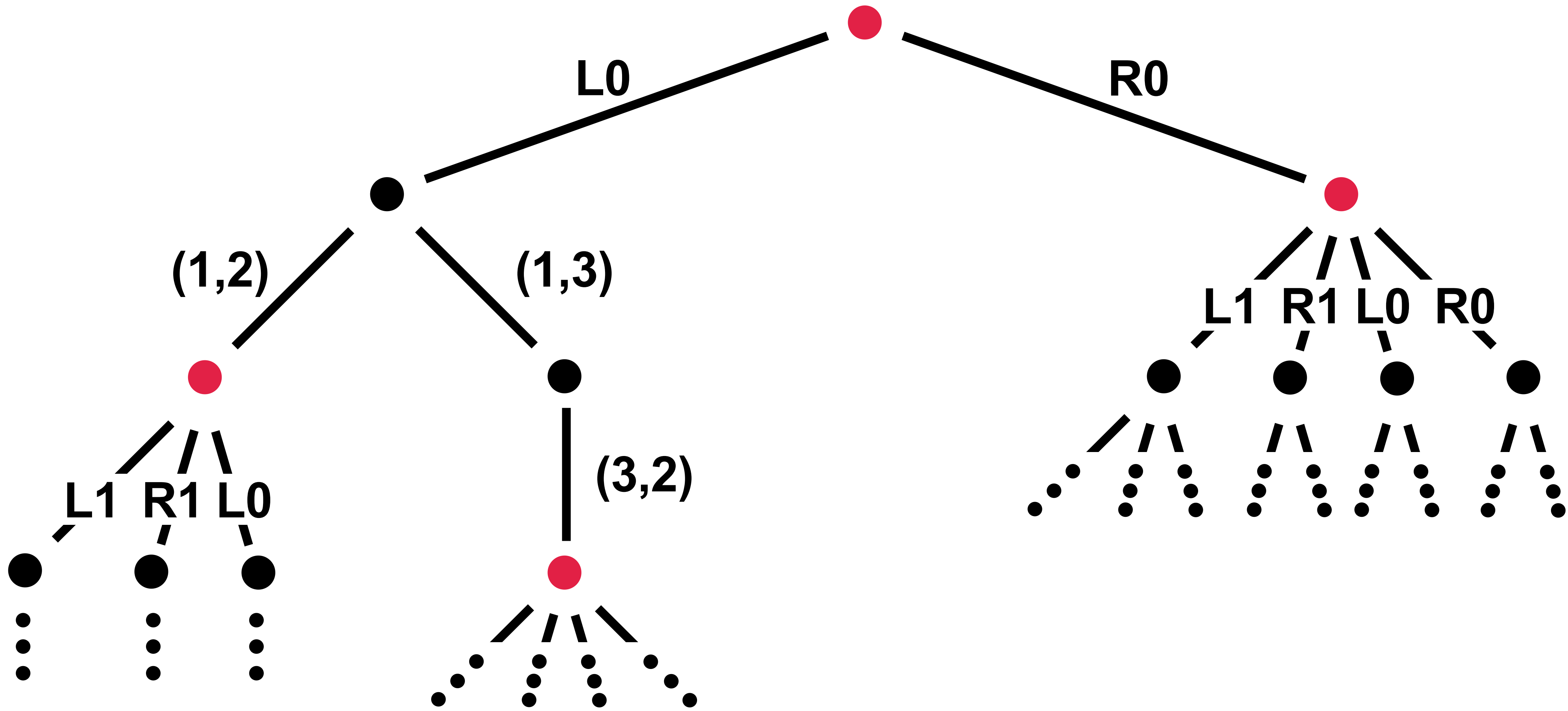
$$2 \cdot 4 \cdot 6 \cdot \dots = 2^q q!$$



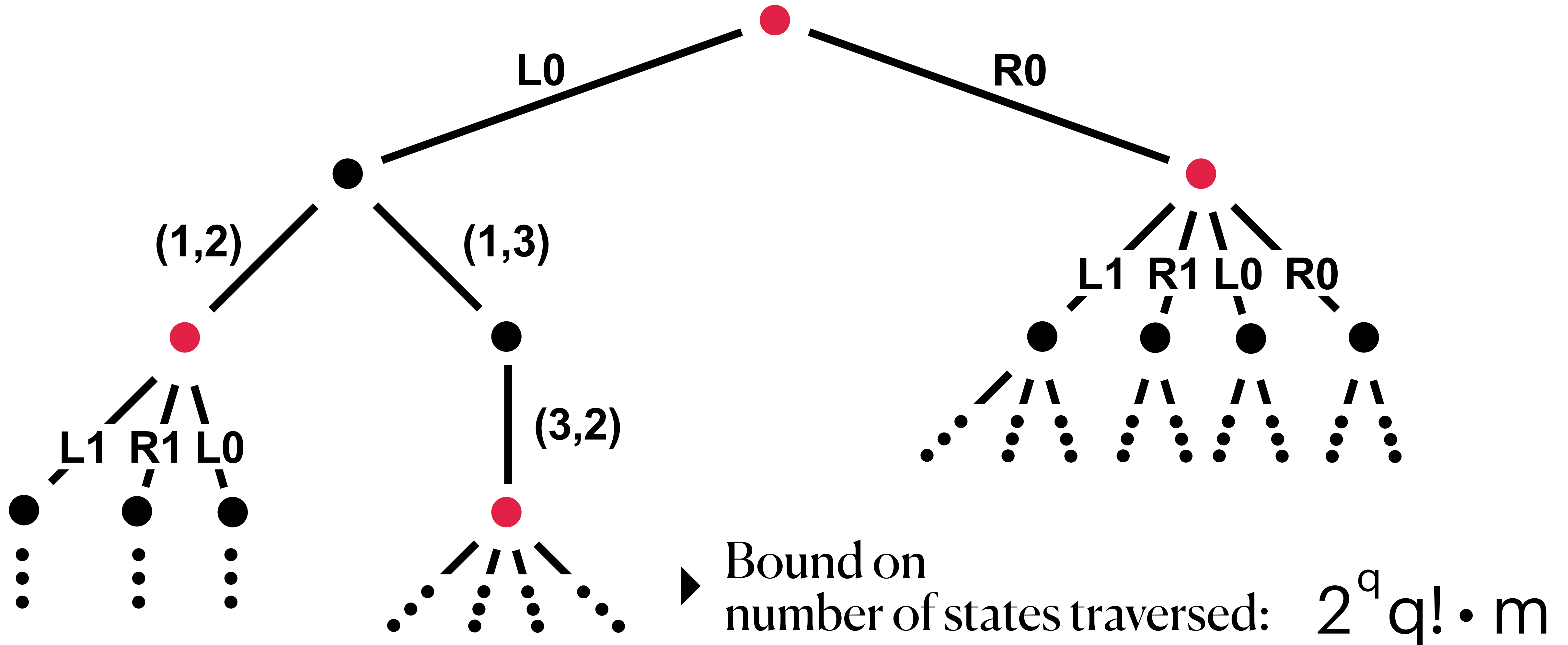
Decision tree \times Labellings



Putting it together

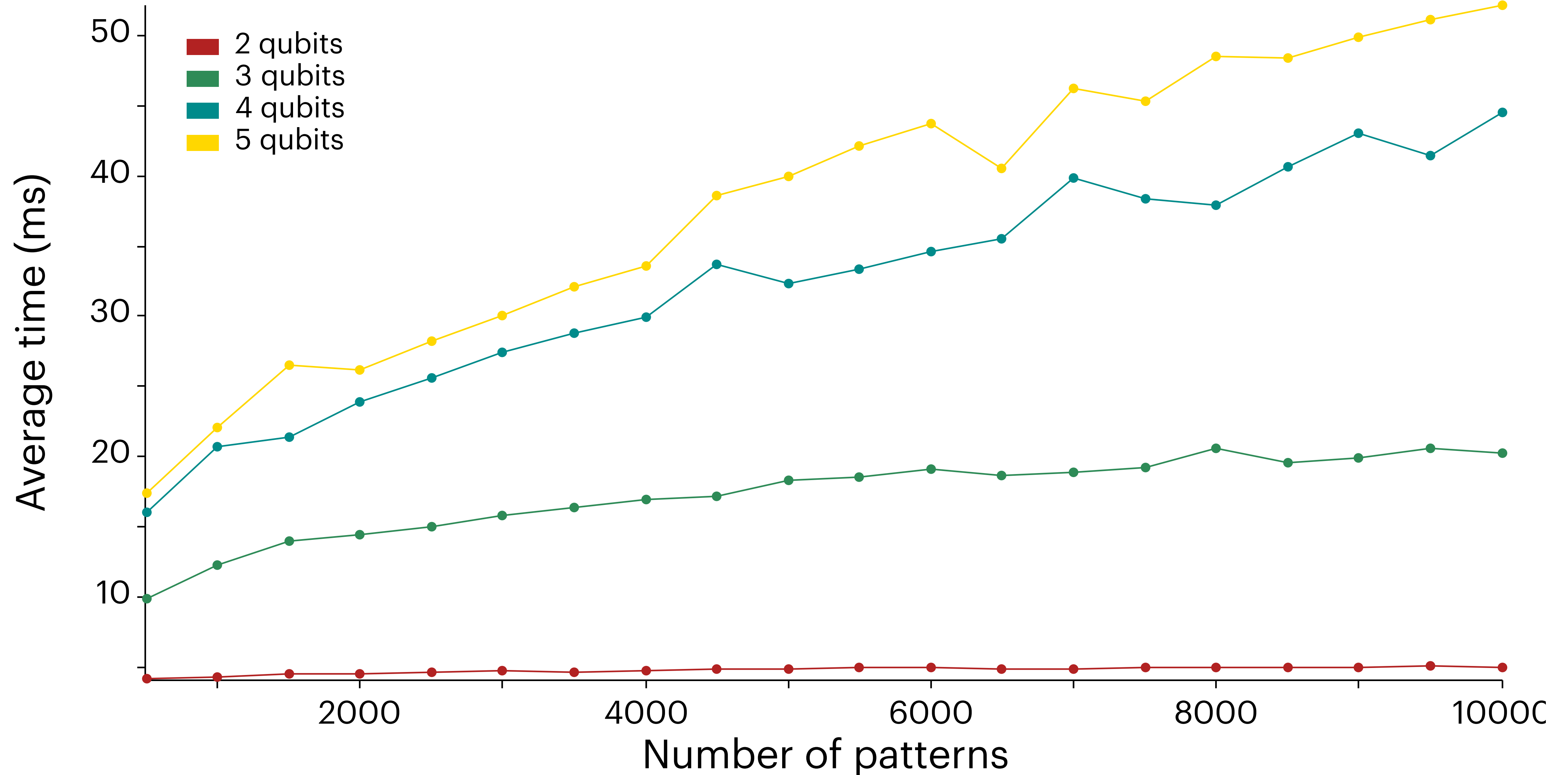


Putting it together

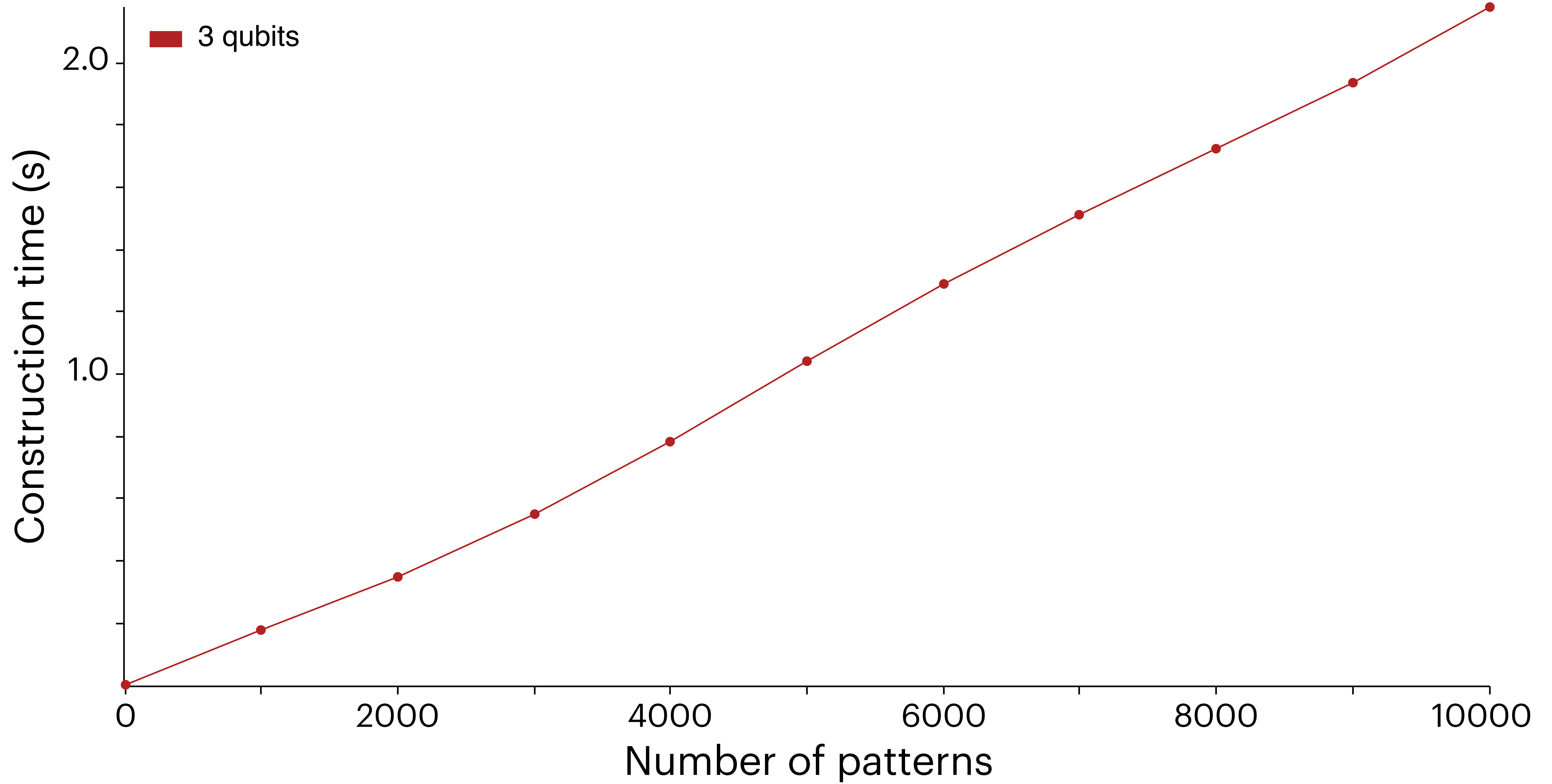


Benchmarks!

Pattern matching times for $n=2000$ input



Automaton construction times



All code available at

`github.com/lmondada/portmatching`

