

量子计算与机器学习 Lab4 Report

PB21111653

李宇哲

第 1 题 编写并实现量子傅里叶变换电路

1-1

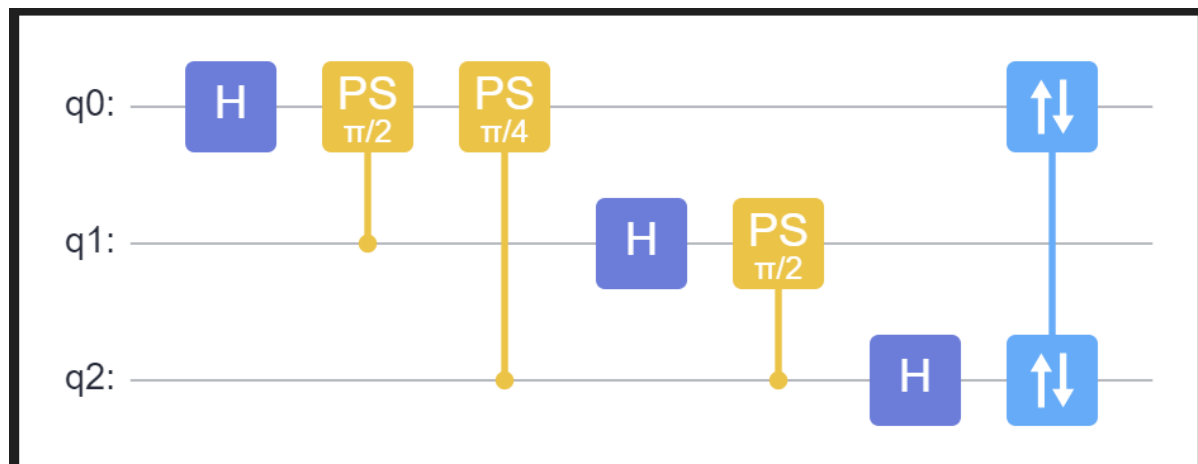
使用3个量子比特，编写一个 QFT 的量子电路

环境依赖

```
1 import numpy as npy
2 from mindquantum.core import H, X, S, T
3 from mindquantum.core import Circuit
4 from mindquantum.simulator import Simulator
5 from mindquantum.core import Measure
6 from mindquantum.algorithm.library import qft
```

量子电路如下

```
1 sim = Simulator('mqvector', 3)
2 qft_circuit = qft([0, 1, 2])
3 qft_circuit.svg()
```



1-2

准备两个简单的输入量子态 $|101\rangle$ 和 $|111\rangle$ 并对其应用 QFT。(建议在实验过程中编写一个量子傅里叶变换的函数)

环境依赖

```
1 from mindquantum.core.circuit import Circuit, SwapParts
2 from mindquantum.core.gates import H, PhaseShift
3 from mindquantum.utils.type_value_check import _check_input_type
```

量子傅里叶变换函数如下

```

1 def _rn(k):
2     return PhaseShift(2 * npy.pi / (2 ** k))
3
4 def _qft_unit(qubits):
5     circuit = Circuit(H.on(qubits[0]))
6     for idx, ctrl_qubit in enumerate(qubits[1:]):
7         circuit += _rn(idx + 2).on(qubits[0], ctrl_qubit)
8     return circuit
9
10 def qft(qubits):
11     _check_input_type('qubits', (list, range), qubits)
12     circuit = Circuit()
13     n_qubits = len(qubits)
14     for i in range(n_qubits):
15         circuit += _qft_unit(qubits[i:])
16     if n_qubits > 1:
17         part1 = []
18         part2 = []
19         for j in range(n_qubits // 2):
20             part1.append(qubits[j])
21             part2.append(qubits[n_qubits - j - 1])
22         circuit += SwapParts(part1, part2)
23     return circuit

```

输入量子态为 $|101\rangle$

```

1 input_state_101 = Circuit()
2 input_state_101 += X.on(0)
3 input_state_101 += X.on(2)
4
5 circuit_101 = input_state_101 + qft_circuit
6 sim.reset()
7 sim.apply_circuit(circuit_101)
8 state_101 = sim.get_qs(ket=True)
9 print("对  $|101\rangle$  状态进行 QFT 变换后的状态: ")
10 print(state_101)

```

对 $|101\rangle$ 状态进行 QFT 变换后的状态:

$$\frac{\sqrt{2}}{4}|000\rangle$$

$$-\frac{\sqrt{2}}{4}|001\rangle$$

$$\frac{\sqrt{2}}{4}j|010\rangle$$

$$-\frac{\sqrt{2}}{4}j|011\rangle$$

$$(-\frac{1}{4}-\frac{1}{4}j)|100\rangle$$

$$(\frac{1}{4}+\frac{1}{4}j)|101\rangle$$

$$(\frac{1}{4}-\frac{1}{4}j)|110\rangle$$

$$(-\frac{1}{4}+\frac{1}{4}j)|111\rangle$$

输入量子态 $|111\rangle$ 状态

```

1 # |111> 状态
2 input_state_111 = Circuit()
3 input_state_111 += X.on(0)
4 input_state_111 += X.on(1)
5 input_state_111 += X.on(2)
6 # 对 |111> 状态进行 QFT 变换
7 circuit_111 = input_state_111 + qft_circuit
8 sim.reset()
9 sim.apply_circuit(circuit_111)
10 state_111 = sim.get_qs(ket=True)
11 print("对 |111> 状态进行 QFT 变换后的状态: ")
12 print(state_111)

```

对 $|111\rangle$ 状态进行 QFT 变换后的状态:

$\sqrt{2}/4|000\rangle$
 $-\sqrt{2}/4|001\rangle$
 $-\sqrt{2}/4j|010\rangle$
 $\sqrt{2}/4j|011\rangle$
 $(1/4-1/4j)|100\rangle$
 $(-1/4+1/4j)|101\rangle$
 $(-1/4-1/4j)|110\rangle$
 $(1/4+1/4j)|111\rangle$

第 2 题 实现 QFT 的逆变换 (IQFT)

2-1

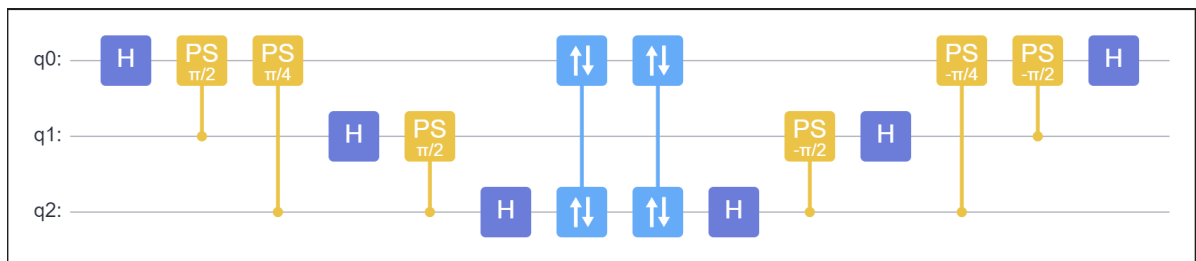
在 QFT 电路后添加 IQFT 电路

IQFT 电路如下

```

1 iqft_circuit = qft_circuit.hermitian()
2 qft_iqft_circuit = qft_circuit + iqft_circuit
3 qft_iqft_circuit.svg()

```



2-2

验证输入态是否可以通过 QFT 和 IQFT 的组合完全恢复

$|101\rangle$

```

1 full_circuit_101 = input_state_101 + qft_iqft_circuit
2
3 sim.reset()
4 sim.apply_circuit(full_circuit_101)
5 final_state_101 = sim.get_qs(ket=True)
6
7 print("对 |101> 状态进行 QFT 变换后再进行逆 QFT 变换后的状态: ")
8 print(final_state_101)

```

对 $|101\rangle$ 状态进行 QFT 变换后再进行逆 QFT 变换后的状态:
 $1|101\rangle$

$|111\rangle$

```

1 full_circuit_111 = input_state_111 + qft_iqft_circuit
2
3 sim.reset()
4 sim.apply_circuit(full_circuit_111)
5 final_state_111 = sim.get_qs(ket=True)
6
7 print("对 |111> 状态进行 QFT 变换后再进行逆 QFT 变换后的状态: ")
8 print(final_state_111)

```

对 $|111\rangle$ 状态进行 QFT 变换后再进行逆 QFT 变换后的状态:
 $1|111\rangle$