技术交底书模板

内部案号：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 发明名称 | 参数化量子电路评估软件（pqc） | | | |
| 发明人信息 | 周青 | | 510105199311201028 | |
| 发明方案关键词 | 量子电路，纠缠能力，表达性 | | | |
| 撰写人 | 周青 | E-mail | | zhouqing@qudoor.cn |
| 部门 | 算法组 | 技术方向 | | 量子算法 |
| 撰写人电话 | 13551806456 |  | |  |
| 专利负责人 | 李李 | E-mail | | lili@qudoor.cn |
| 电话 |  |  | |  |

# 背景技术内容（现有技术）

提供参数化量子电路评估软件的使用说明，介绍该软件的算法实现方式，包括评估指标的定义，作用。让软件使用者了解其对任意电路的使用方式，并解读软件输出结果的含义。

参数化量子电路在经典量子混合（HQC）算法的应用中起到了至关重要的作用。然而，目前为止我们并没有一种行之有效的方式，在电路复杂度一定的前提下来选择能够很好表征特征空间的电路模版。更通俗的来说，量子电路在处理各类基于HQC算法的问题中是一种“可调的参数”，然而我们缺少一种手段让这些“参数”收敛到最优值。这是我们在嘈杂中型量子（NISQ）时代发展HQC算法遇到的一个挑战。

一个攻克这样的技术瓶颈的方法，则是定义一类能够有效描述量子电路的指标，或者说--品质因数。通过这类品质因数我们能够更直观的在具体解决某一类算法问题时量化量子电路的好坏。本评估软件则是在这样的前提下设计出来达到评估量子电路在针对解决某一种问题时好坏的一种工具。

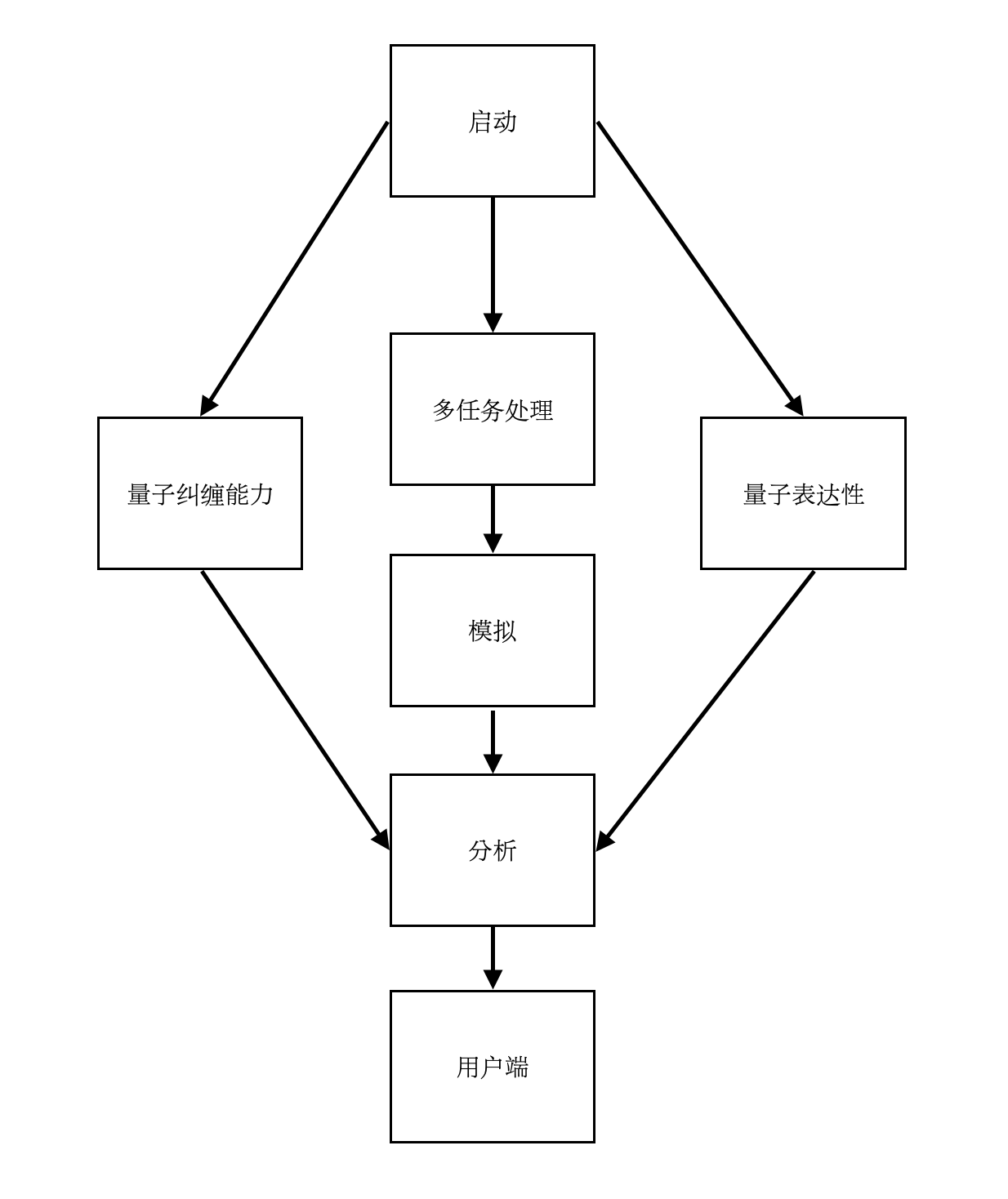
# 2、本发明技术方案的详细内容（本技术方案）

# 2软件设计

## 2.1软件概括

本软件针对由用户自主创建的参数化量子电路，通过量子纠缠能力（Q）和量子表达性（E）两个品质因数的输出值，来评估所测试电路的好坏。

## 2.2软件系统结构



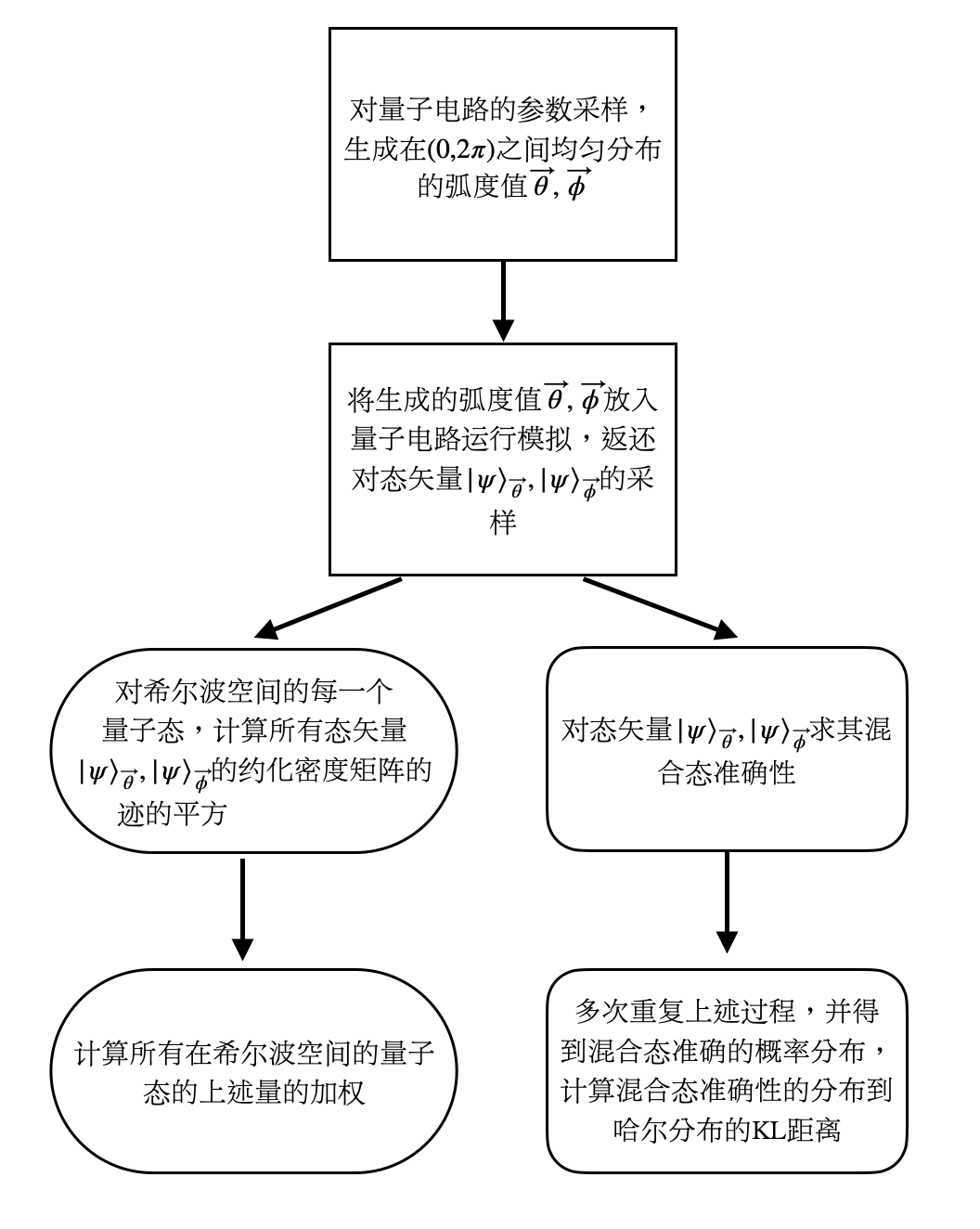
*图一、评估软件系统结构图*

软件系统结构如图一所示，主要由分布于四层的七大模块构成：最底层为启动层，主要作用为读出用户输入的量子电路的参数。启动模块上端链接多任务处理，量子纠缠能力和量子表达性三大模块。其中多任务处理模块的主要功能在于创建一个任务池，实现任务在CPU上的告诉并行功能。该模块直接链接模拟模块，其功能在于对模拟电路运行工作的定义，并通过调用多任务模块实现电路模拟的高效并行运算。其余两大模块分别承载对于量子纠缠能力和量子表达性的实现的核心算法部分。第三层为分析层，由分析模块组成。分析模块会根据最顶层用户端定义的需求：即：

1. 分析量子电路的量子纠缠能力；或
2. 分析量子电路的量子表达性；或
3. 分析量子电路的量子纠缠能力和量子表达性

来决定调用哪几个和需求相对应的模块。最顶层为用户层的用户端，主要功能为录入用户定义的量子电路，用户需求的量子电路品质因数，评估品质因数的方法，等等。

## 2.3软件核心算法

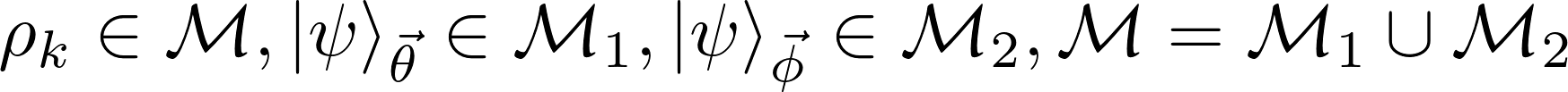
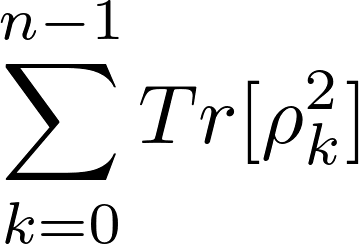


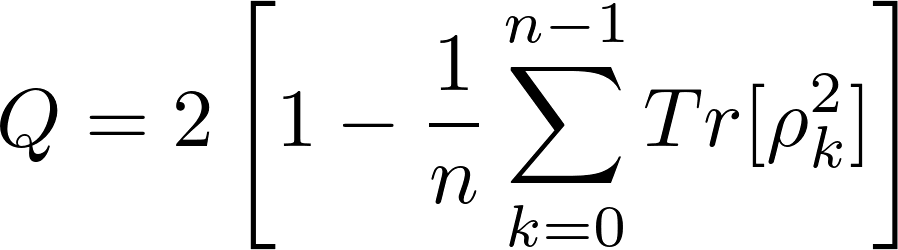
*图二、软件品质因数算法流程图*

软件品质因数算法流程图如图二所示：其中底层（方形模块）为两个品质因数的计算中共同部分，椭圆模块所在分支为计算量子纠缠能力所独有；方圆模块所在分支为计算量子表达性所独有。用户定义的量子电路会首先被分析其参数化门，然后对所有门的参数进行两组多次采样。采样后的弧度值会被分别放回到量子电路，进行多线程模拟。模拟后的态矢量会分别被两个计算分支使用。

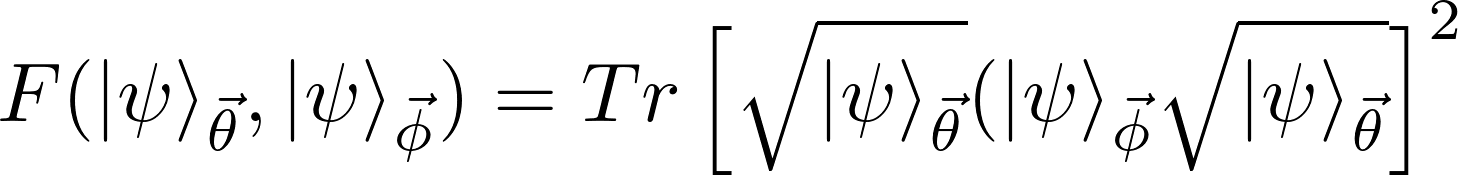
在计算量子纠缠能力时，我们可以根据以下步骤得到品质因数Q：

1. 首先对每个量子比特k，计算其量子态矢量的约化密度矩阵的迹的平方（纯度）：

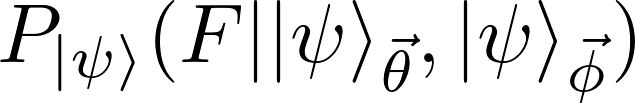
其中，

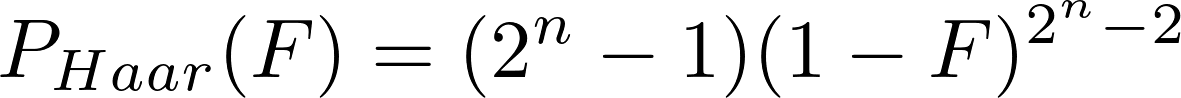
****2. 然后，对所有在希尔波空间的量子态的纯度进行累加：

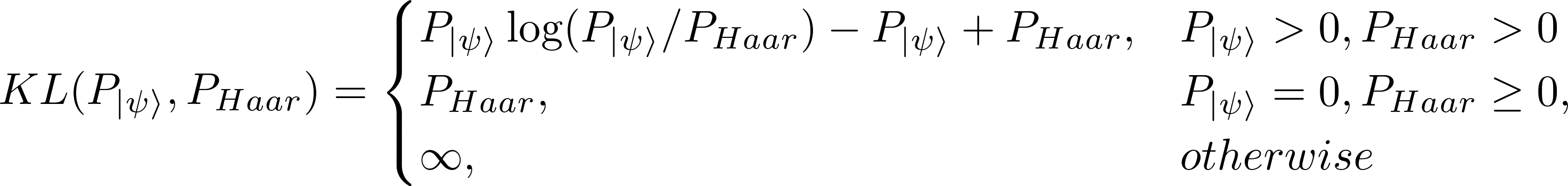
3. 最后，计算品质因数

在计算量子表达性时，我们可以根据以下步骤得到品质因数E：

1. 计算态矢量的混合态准确性：

2. 然后，多次测量混合态准确性，并得到混合态准确性的分布：

3. 最后，将上述分布于哈尔分布进行比较, 并计算上述分布到哈尔分布的KL距离：

****

# 3运行环境

## 3.1 Python版本

3.9及以上

## 3.2 Python包

numpy>=1.21.3

scipy>=1.7.1

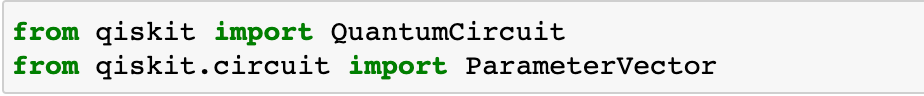
qiskit=0.18.3

# 4使用说明

## 4.1调用qiskit的函数

我们可以通过调用python包“qiskit”，来创建一个参数化量子电路。具体操作方式如下：

1. 从qiskit包中调用QuantumCircuit，用来承载用户自定义的量子电路
2. 从qiskit包中调用ParameterVector，用来承载放入量子电路里的参数

如下图所示：

*图三、调用qiskit示意图*

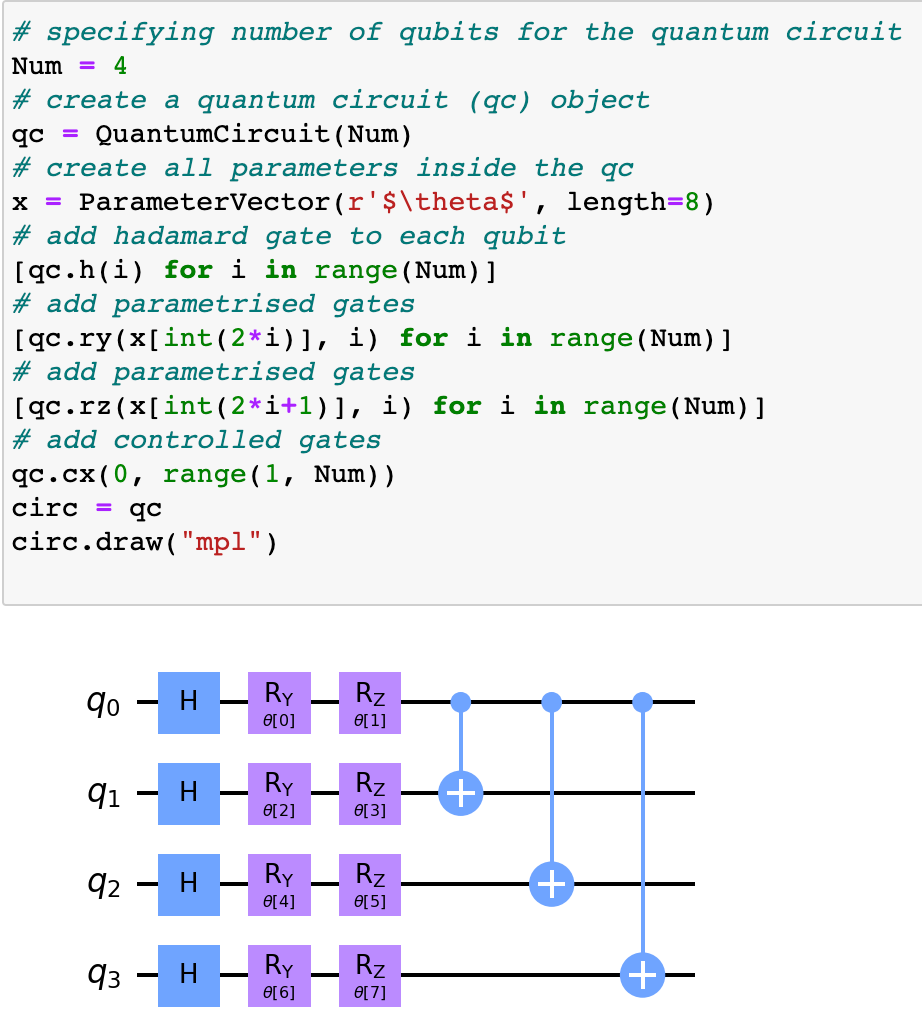
## 4.2创建一个量子电路

1. 首先需要申明量子电路的比特数，如图四所示，我们创建一个4比特的量子电路
2. 创建包含量子电路所有参数的序列，通过ParameterVector来实现

3.在量子电路的每个比特后面接上一个hardamard门，此门的功效为使每个比特产生混合态（superposition）

4.将创建的参数通过参数化量子门，如rx，ry，rz，的方式，放进量子电路5.用控制门实现

比特与比特之间的纠缠（entanglement）

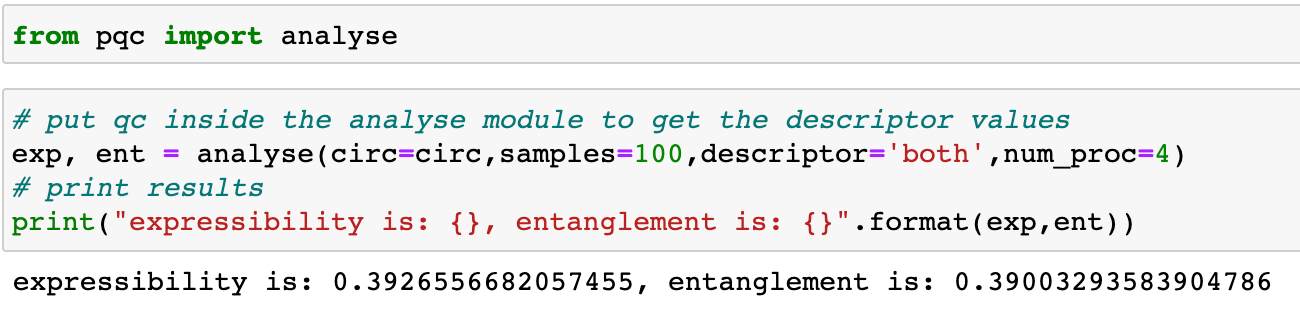
1. 通过qc.draw画出刚刚创建的量子电路

如下图所示：

*图四、创建量子电路示意图*

## 4.3将量子电路放入软件并输出结果

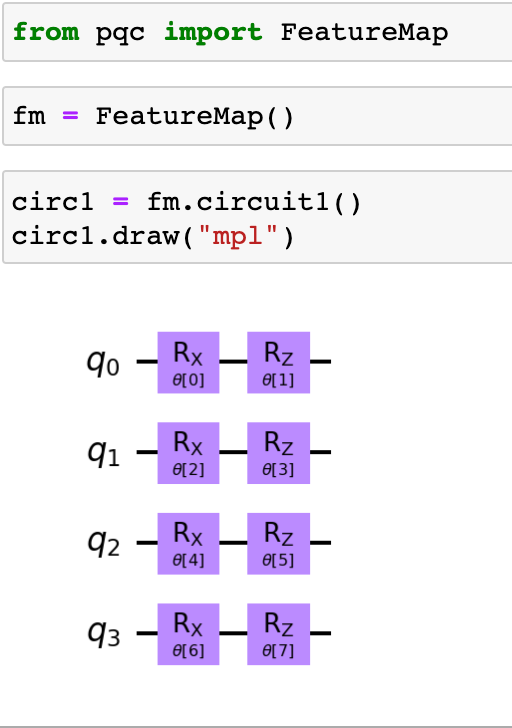
我们可以通过调用软件包pqc中的analyse函数，来实现对所创建的量子电路进行品质因数的评估分析，其返还值根据用户指定的品质因数的不同，会有区别。当“descriptor”被指定为“ex”的时候，返还值为一个浮点数，即该量子电路的E；当“descriptor”被指定为“ec”的时候，返还值为一个浮点数，即该量子电路的Q；当“descriptor”被指定为“both”的时候，两个品质因数的测量值都会被返还，如下图所示：

*图五、输出结果示意图*

# 5实例展示

除了上述实例之外，我们还提供了一套量子电路库，可供用户参考学习。调用库内自带量子电路的方式为：

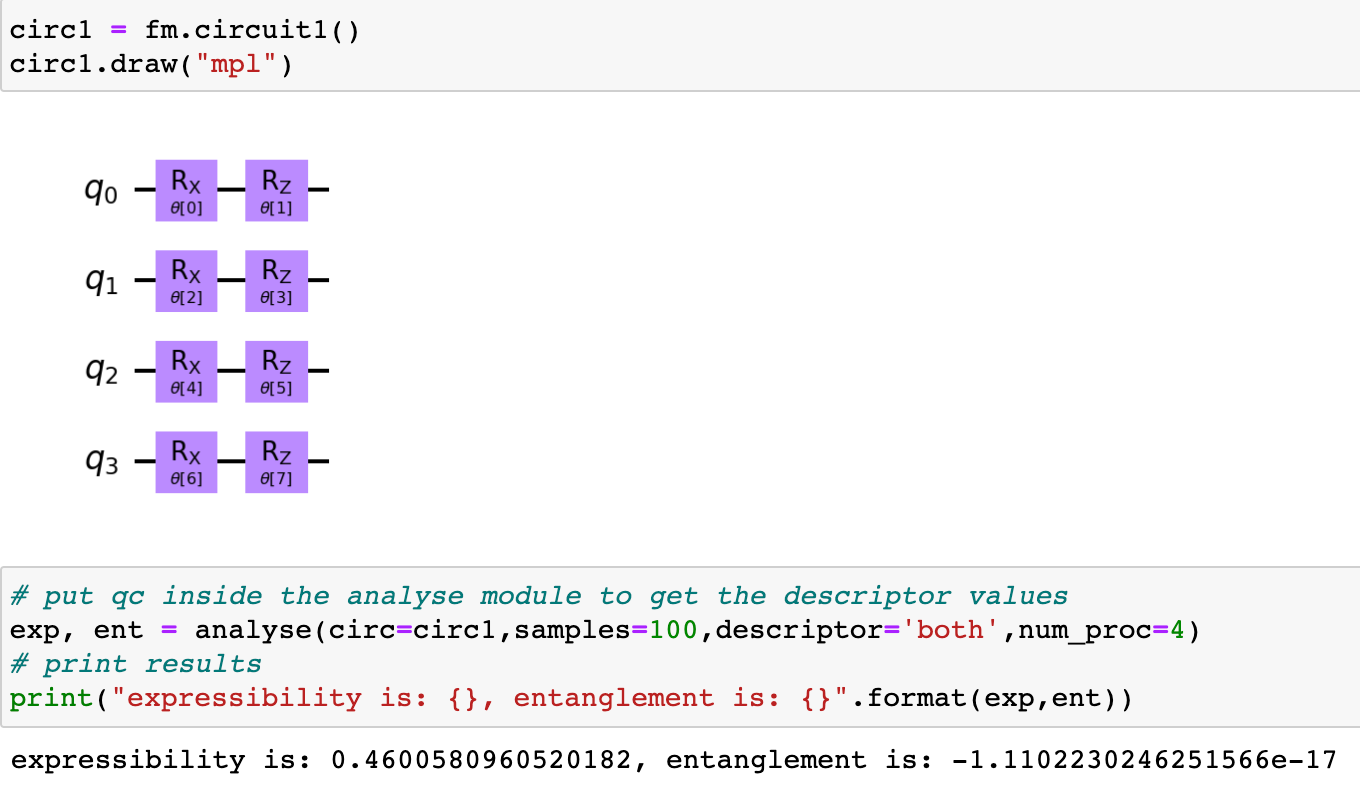
1. 首先，从pqc包中调出FeatureMap函数，这即是包含pqc自带的所有量子电路的库
2. 其次，从库中取出一个量子电路。目前库中有11个量子电路，命名规则为circuit+序号，序号值为从1到11之间的任意整数。

如下图所示：

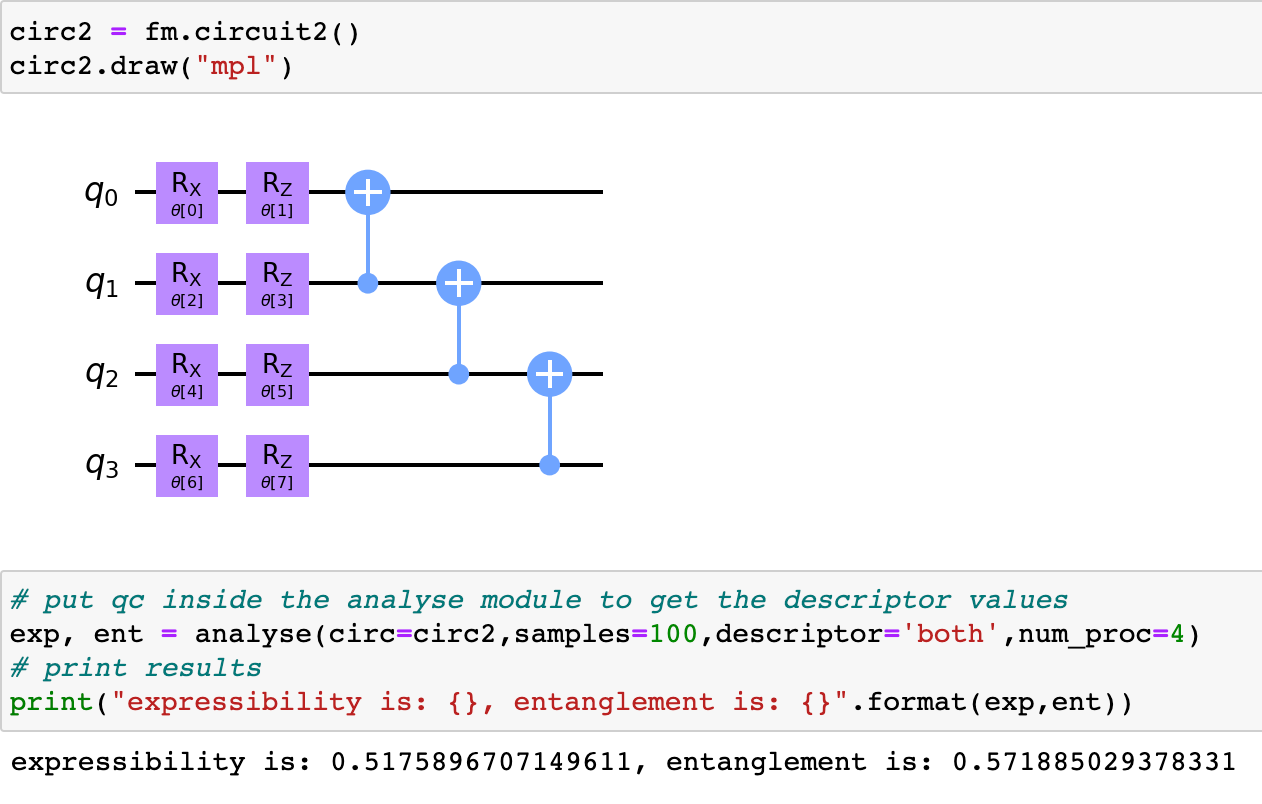
*图六、取出库中自带量子电路*

## 5.1实例一

如图七所示，我们可以通过pqc包中的analyse函数直接分析从库中取出的一号量子电路的品质因数。

*图七、**一号量子电路的两个品质因数输出结果*

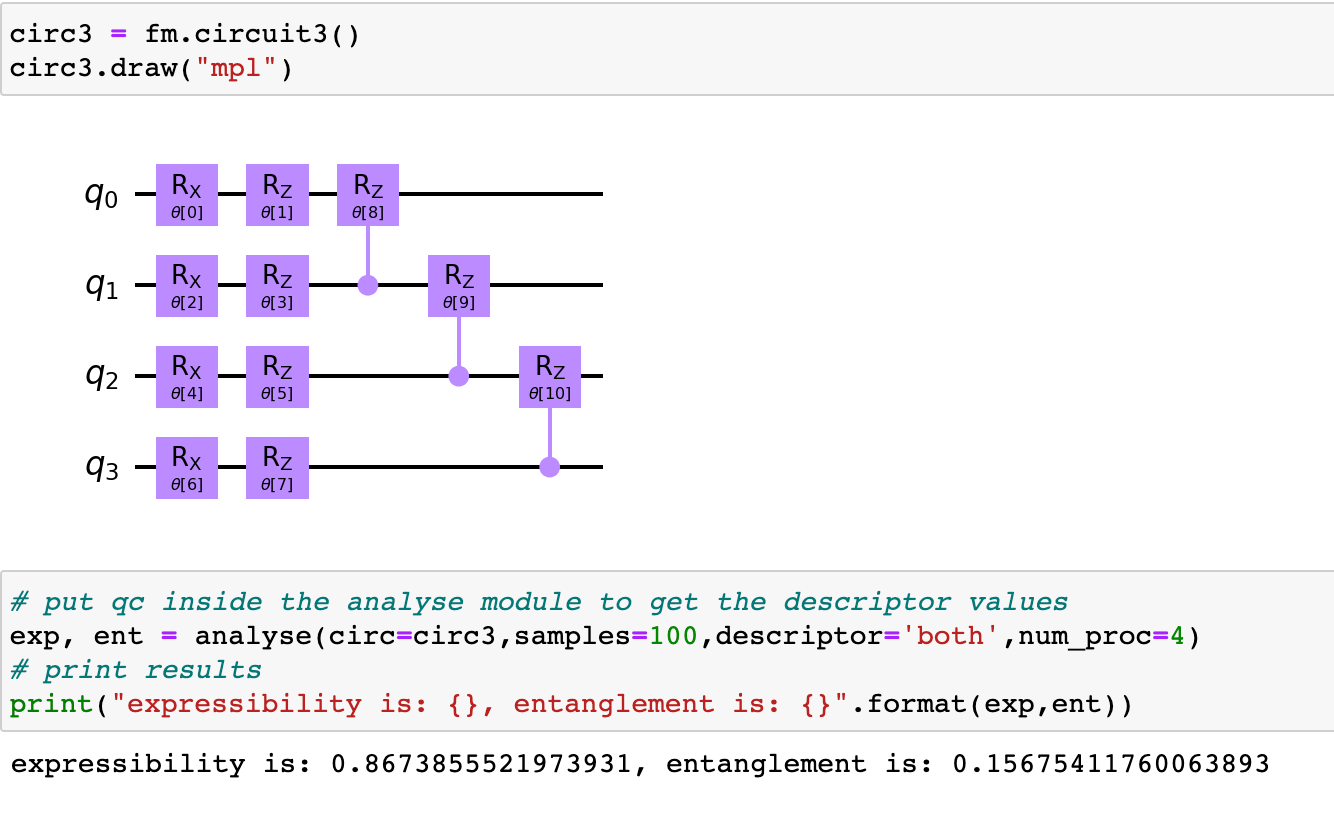
## 5.2实例二

如图八所示，我们可以通过pqc包中的analyse函数直接分析从库中取出的二号量子电路的品质因数。

*图八、二号量子电路的两个品质因数输出结果*

## 5.3实例三

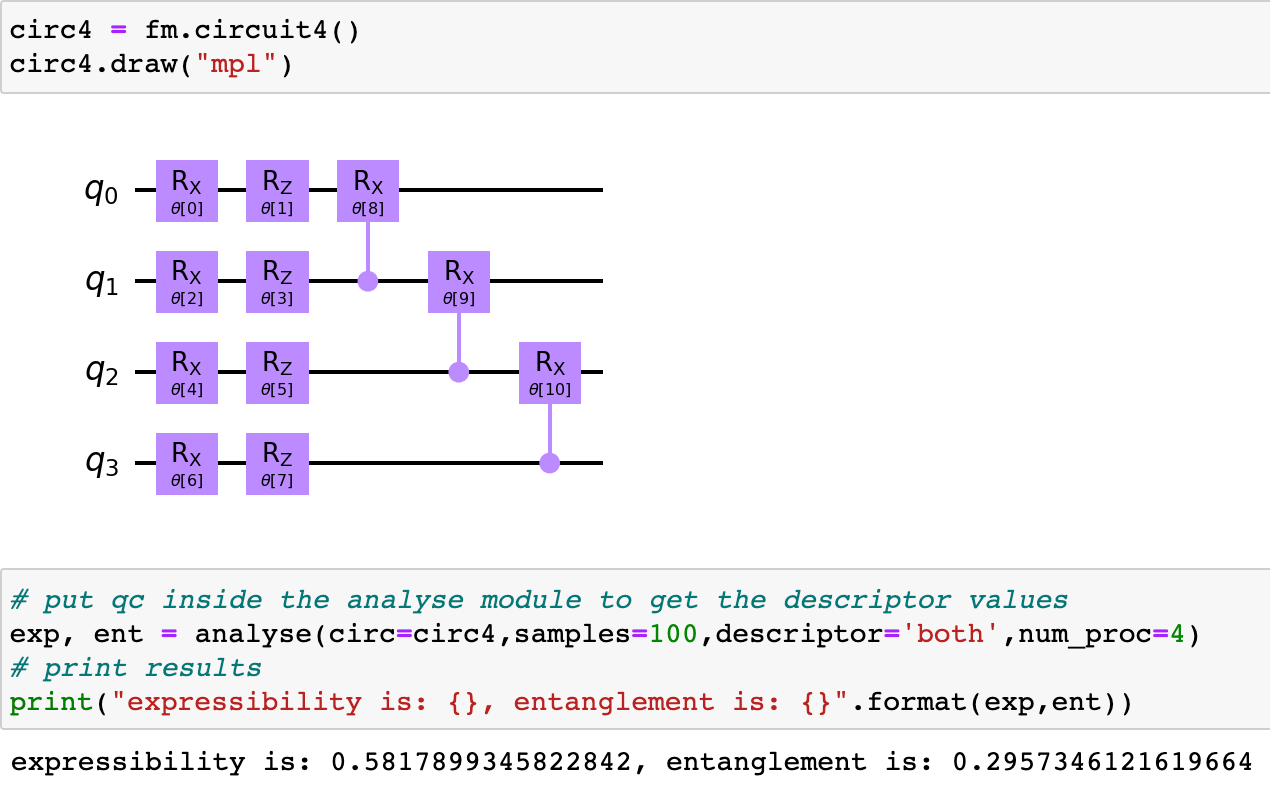
如图九所示，我们可以通过pqc包中的analyse函数直接分析从库中取出的三号量子电路的品质因数。



*图九、三号量子电路的两个品质因数输出结果*

## 5.4实例四

如图十所示，我们可以通过pqc包中的analyse函数直接分析从库中取出的四号量子电路的品质因数。

*图**十、四号量子电路的两个品质因数输出结果*

# 3、针对2中的技术方案的内容，是否还有别的替代方案

1、替代方案可以是部分结构、器件、方法步骤的替代，也可以是完整技术方案的替代，以能够实现被替代技术的目的、效果为准；

2、替代方案可以完全重写，也可以针对被替代的部分，写出其中的差别。

# 4、本发明技术方案的关键点

1、也就是你认为哪些地方与现有技术的不同或创新；

2、根据你的理解，对这些关键点进行重要性的排序。

附件：

参考文献（如专利/论文/标准）