Compiler Optimization Formalism

1 Notation and Problem Statement

Given a quantum program, we wish to compile selectively to target parallelism and code size. Specifically, we wish to extract as much parallelism as possible while retaining the smallest possible code size.

Definition 1.1. A *Primitive Gate Operation* is an element of the set $\mathcal{G} = \{H, X, Z, CNOT, ...\}$. These are elementary quantum operations chosen from any particular set of universal operations. Shown here is one particular choice.

Definition 1.2. A *Function* is an ordered list of operations $(p_1, p_2, ...p_n)$. Each p_i is an *Operation*, which is a member of one of two sets: $\mathcal{F} = \{\alpha, \beta, \gamma...\}$ a list of functions, or $\mathcal{G} = \{H, X, CNOT, T, ...\}$ the set of primitive gate operations. The set of all *Operations* $\mathcal{O} = \mathcal{F} \cup \mathcal{G}$.

Definition 1.3. An input program P is itself a *Function*, with a corresponding $P_\mathcal{F}$, a list of functions used within the program. Each element $p_i \in P$ is a member of either $\mathcal{G}$ or $P_\mathcal{F}$. Each element of the set $P_\mathcal{F}$ is a function, comprised itself of elements of either $\mathcal{G}$ or $P_\mathcal{F}$.

Definition 1.4. Given an input program P, *Code Size* (CS) of P is defined as:

$$CS_P = |P| + \sum_{\alpha \in P_\mathcal{F}} |\alpha| \tag{1}$$

This definition captures code size in the unit of instructions: each element of P is considered to be a single instruction, and every element of $P_\mathcal{F}$ is considered as a list of instructions. The total code size is the number of instructions in the main input program P plus the total size of all of the functions called within the program.

Definition 1.5. Associated with each $p_i \in P$ is a corresponding list DEP_{p_i}, defined as a list of operations upon which p_i is *dependent*. The list DEP_{p_i} is a list of operations $p_j \in P$, where $j < i$. Indices are considered as operation steps in the original, sequential program P.

Definition 1.6. Also associated with each $p_i \in P$ is an integer v_{p_i}. This denotes the *Parallelism* of p_i, defined with the following procedure:

$f \leftarrow \max_j p_j \in DEP_{p_i}$

$v_{p_i} \leftarrow 0$

For each $p_k \in P$ such that $k \in [i, f]$:

If Type(p_k) = Type(p_i), then increment v_{p_i}

2 Objective Function

3 Linear Program Formalism and Transformation