A Symbolic Approach to Quantum Computation Simulation*

António Pereira and Rosália Rodrigues

Department of Mathematics,
University of Aveiro -Portugal,
{antoniop, rosalia}@mat.ua.pt

Abstract. This paper presents SQCS: a *Mathematica* package for the symbolic simulation of quantum computation. It can be a useful tool in the development and testing of quantum algorithms as well as in the teaching and learning of quantum computation.

The strength of this symbolic approach is here illustrated with Grover's quantum algorithm for database search.

1 Introduction

Simulating quantum computations on a classical computer is, in general, a hard task. Every classical simulator of the dynamics of a quantum computer faces two problems: the representation of quantum states and its unitary evolution. (See, for example, Nielsen and Chuang [1] or Kitaev [2] for introductory readings on the basic concepts of quantum computation.)

In fact, a pure state of a composite closed quantum system with n d-level qudits can be seen as a unit vector on a Hilbert space of dimension d^n, i.e., a normalized vector with d^n complex components. Also every unitary operator defined in a Hilbert space of dimension d^n can be described by a $d^n \times d^n$ complex unitary matrix. Working directly with these vectors and matrices becomes unfeasible even when the number of qudits is small.

Most of the existing quantum computation simulators follow the linear algebra matrix approach, ranging from simple applets that handle computations with a maximum of 4 qudits to heavy clusters of computers capable of solving problems with a maximum of 30 qubits.

The complexity of simulating the state evolution of a quantum computer, for a given quantum algorithm, is strongly related with the number of entangled qudits in the system. We expect a quantum algorithm to admit an efficient simulation when, along the simulation process, only small subsystems of the whole system become entangled.

* This work was partially supported by the R&D unit Center for Research in Optimization and Control (CEOC) of the University of Aveiro – Portugal.

© Springer-Verlag Berlin Heidelberg 2006

2 The Symbolic Quantum Computation Simulator

On trying to build a simulator over an environment like *Mathematica*[3], the first observation to take into account is that not the content of a quantum object need to be stored, but only its identifier. To our knowledge, even the existing quantum computation applications developed with *Mathematica* fail to take advantage of this fact.

For instance, any basis state of a qudit on a d-dimensional Hilbert space need not be represented as a full length d list, not even as a packed array, but simply as a symbolic labelled object with head `ket`. Subsequent operations on such objects should work only symbolically. Also every unitary operator can be represented by a list of rules that define its symbolic action on quantum states.

The basic objects and operators defined in SQCS are introduced next. Most have external notations for the *Mahematica* frontend associated with the internal representations.

2.1 Kets

For $d \geq 2$ let $\mathbb{Z}_d$ denote the set $\{0, \ldots, d-1\}$. A *qudit* is a quantum system whose state is a unit vector in a d-dimensional Hilbert space. We assume that the underlying Hilbert space is always $\mathcal{H}_d = \mathbb{C}^d$. It has become common practice to identify the canonical computational basis of $\mathcal{H}_d$ with the set $\{|k\rangle, k \in \mathbb{Z}_d\}$, in Dirac notation.

The general state of a qudit is $|\psi\rangle = \sum_{k\in\mathbb{Z}_d} a_k |k\rangle$, where $a_k \in \mathbb{C}$, for $k \in \mathbb{Z}_d$ and $\sum_{k\in\mathbb{Z}_d} |a_k|^2 = 1$. The case $d = 2$ gives the usual definition of a *qubit* with general state, $|\psi\rangle = a|0\rangle + b|1\rangle$, $|a|^2 + |b|^2 = 1$, $a, b \in \mathbb{C}$.

In SQCS each basis state of a Hilbert space is represented by a symbolic object with head `ket`. Figure 1 shows some examples and basic properties of `ket` objects.

```
<< SQCS`

{ket[0], ket[1], ket[0, 3], ket[1, 3], ket[2, 3]}

{| 0⟩, | 1⟩, | 0₃⟩, | 1₃⟩, | 2₃⟩}

α | ψ⟩ + α | ϕ⟩ // Simplify

α (| ϕ⟩ + | ψ⟩)

α | ϕ⟩ + β | ϕ⟩ // Simplify

(α + β) | ϕ⟩

| ϕ⟩ - | ϕ⟩

0
```

```
<< SQCS`

| ψ₀⟩ = 1 / Sqrt[2] | 0⟩ + I / Sqrt[2] | 1⟩
```

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

```
| ψ₁⟩ = I | ψ₀⟩ + 1 / Sqrt[2] | 1⟩
```

$$\frac{|1\rangle}{\sqrt{2}} + i\left(\frac{|0\rangle}{\sqrt{2}} + \frac{i\,|1\rangle}{\sqrt{2}}\right)$$

```
| ψ₁⟩ // Simplify
```

$$\frac{i\,|0\rangle}{\sqrt{2}}$$

Fig. 1. Some algebraic properties of ket objects

2.2 Bras

The set of linear functionals on a Hilbert space $\mathcal{H}$ is also an Hilbert space, called the dual space of $\mathcal{H}$ and denoted by $\mathcal{H}^\dagger$. According to Riesz theorem, the inner product in $\mathcal{H}$ induces a one to one correspondence between the elements of $\mathcal{H}$ and $\mathcal{H}^\dagger$. In Dirac notation, the dual of the ket state $|\psi\rangle \in \mathcal{H}$ is the *bra* state $\langle\psi| = |\psi\rangle^\dagger \in \mathcal{H}^\dagger$.

In SQCS the dual of a `ket` object is another object with head `bra`. Figure 2 displays some basic properties of `bra` objects and of the `dagger` function.

2.3 BraKets and the Inner Product

Considerer a `bra` object $\langle\psi| \in \mathcal{H}^\dagger$. Since $\langle\psi|$ is a linear functional in $\mathcal{H}$, the action of $\langle\psi|$ on a `ket` object $|\phi\rangle \in \mathcal{H}$, $\langle\psi|\,(|\phi\rangle)$ gives a complex number, denoted by the `braKet` $\langle\psi|\phi\rangle$.

Objects of type `braKet` can also have a different meaning. Recall that the Hilbert space $\mathcal{H}$ is equipped with a inner product, assumed to be conjugate-linear in the first argument and linear in the second. If we denote the inner product of $|\psi\rangle$ and $|\phi\rangle$ by $\langle\psi|\cdot|\phi\rangle$, then $\langle\psi|\cdot|\phi\rangle = \langle\psi|\phi\rangle = \langle\psi|\,(|\phi\rangle)$.

Some properties of the inner product of kets are shown in fig. 3.

2.4 The Kronecker Product

The underlying Hilbert space of a composite system, say a system composed of two qubits, is the tensor product space of the underlying Hilbert spaces of the

```
<< SQCS`

{bra[0], bra[1], bra[0, 3], bra[1, 3], bra[2, 3]}
```

$$\{\langle 0\,|,\ \langle 1\,|,\ \langle 0_3\,|,\ \langle 1_3\,|,\ \langle 2_3\,|\}$$

$$(\alpha\,|\,\psi\rangle + \beta\,|\,\phi\rangle)^\dagger$$

$$\langle\psi\,|\,\alpha^* + \langle\phi\,|\,\beta^*$$

```
<< SQCS`
```

$$|\,\psi_0\rangle = 1\,/\,\mathtt{Sqrt[2]}\ |\,0\rangle + \mathtt{I}\,/\,\mathtt{Sqrt[2]}\ |\,1\rangle$$

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

$$|\,\psi_0\rangle^\dagger$$

$$\frac{\langle 0\,|}{\sqrt{2}} - \frac{\mathrm{i}\,\langle 1\,|}{\sqrt{2}}$$

Fig. 2. Some algebraic properties of bra objects

```
<< SQCS`
```

$$\{\langle 0\,|\,1\rangle,\ \langle 1\,|\,1\rangle,\ \langle 1_3\,|\,2_3\rangle\}$$

```
{0, 1, 0}
```

$$\{\langle 0\,|\cdot|\,1\rangle,\ \langle 1\,|\cdot|\,1\rangle,\ \langle 1_3\,|\cdot|\,2_3\rangle\}$$

```
{0, 1, 0}
```

```
<< SQCS`
```

$$(\alpha\,\langle\psi\,| + \beta\,\langle\phi\,|)\cdot|\,\eta\rangle\ \mathtt{//.\ expandInnerProduct}$$

$$\langle\psi\,|\,\eta\rangle\,\alpha^* + \langle\phi\,|\,\eta\rangle\,\beta^*$$

$$\langle\eta\,|\cdot(\alpha\,|\,\phi\rangle + \beta\,|\,\psi\rangle)\ \mathtt{//.\ expandInnerProduct}$$

$$\alpha\,\langle\eta\,|\,\phi\rangle + \beta\,\langle\eta\,|\,\psi\rangle$$

Fig. 3. Some algebraic properties of the inner product of kets

```
<< SQCS`

| φ⟩ ⊗ (| ψ⟩ ⊗ | η⟩) === (| φ⟩ ⊗ | ψ⟩) ⊗ | η⟩

True

(α | φ⟩) ⊗ | ψ⟩

α | φ⟩ ⊗ | ψ⟩

| φ⟩ ⊗ (α | ψ⟩)

α | φ⟩ ⊗ | ψ⟩
```

```
<< SQCS`

(α | φ⟩ + β | ψ⟩) ⊗ | η⟩ //. expandKron

α | φ⟩ ⊗ | η⟩ + β | ψ⟩ ⊗ | η⟩

| η⟩ ⊗ (α | φ⟩ + β | ψ⟩) //. expandKron

α | η⟩ ⊗ | φ⟩ + β | η⟩ ⊗ | ψ⟩

(| 0⟩ + | 1⟩)^⊗2 /. expandPow //. expandKron

| 0⟩ ⊗ | 0⟩ + | 0⟩ ⊗ | 1⟩ + | 1⟩ ⊗ | 0⟩ + | 1⟩ ⊗ | 1⟩
```

Fig. 4. Some algebraic properties of the tensor product

subsystems, $\mathcal{H}_2^{\otimes 2} = \mathcal{H}_2 \otimes \mathcal{H}_2$. Thus the state of a quantum system with n qudits is a unit vector of ${\mathcal{H}_d}^{\otimes n}$. This space has a computational basis with d^n states denoted by $|x_{n-1}\rangle \otimes \cdots \otimes |x_1\rangle \otimes |x_0\rangle$, where $x_j \in \mathbb{Z}_d$, $j = 0, 1, \ldots, n-1$.

Each basis state in ${\mathcal{H}_d}^{\otimes n}$ naturally associates with the radix d representation of the integer $\sum_{j=0}^{n-1} x_j d^j \in \mathbb{Z}_{d^n}$, hence the following notations are equivalent: $|x_{n-1}\rangle \otimes \cdots \otimes |x_1\rangle \otimes |x_0\rangle \equiv |x_{n-1} \cdots x_1 x_0\rangle \equiv \left|\sum_{j=0}^{n-1} x_j d^j\right\rangle$.

A state $|\psi\rangle \in {\mathcal{H}_d}^{\otimes n}$ is called a *product state* if it can be decomposed as $|\psi\rangle = |\psi_{n-1}\rangle \otimes \cdots \otimes |\psi_1\rangle \otimes |\psi_0\rangle$, for some sequence of states $|\psi_i\rangle \in \mathcal{H}_d$, $i = 0, \ldots, n-1$. If no such sequence exists, $|\psi\rangle$ is said to be an *entangled state*.

The tensor product, also known as the Kronecker product, is associative, noncommutative and distributive with respect to linear combinations. Figure 4 displays some examples.

2.5 Operators

In SQCS every linear operator is represented by an object `op[name_,n_,f_]`, where `name` is a label for the operator, `n` is the number of qudits on which the operator acts, and `f` is the function that defines the action of the operator on the basis qudits.

The efficient simulation of the action of a unitary operator on a given quantum state is directly related to the possibility of decomposing it as a tensor product of operators acting on a small number of qudits. Thus a major concern when defining a new operator in SQCS is to represent it as factored as possible.

Several operators are already available in SQCS. In particular, the identity operator I_n, where n is the number of qudits on which the operator acts (thus diverging from the usual notation where subscripts represent the dimension of the underlying Hilbert space of the operator). Other examples of provided operators are:

The Hadamard Operator. One of the fundamental operators in quantum computation for creating superpositions of states is the Hadamard operator, H. It acts on one qubit by $\mathrm{H}\,|i\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^i\,|1\rangle\right)$, for $i = 0, 1$. Figure 5 illustrates the fact that H is its own inverse.

```
<< SQCS`

numericOff

H · |0⟩

(|0⟩ + |1⟩)/√2

H · |1⟩

(|0⟩ - |1⟩)/√2
```

```
<< SQCS`

numericOff

H · (|0⟩ + |1⟩)/√2 //. expandLinearInside // Simplify

|0⟩

H · H

I₁
```

Fig. 5. The Hadamard operator in action

```
<< SQCS`

numericOff;

H^⊗4 · |0⟩^⊗4

1/4 (|0⟩ + |1⟩)^⊗4

H^⊗3 · |0⟩⊗|1⟩⊗|0⟩

((|0⟩ + |1⟩) ⊗ (|0⟩ - |1⟩) ⊗ (|0⟩ + |1⟩))/(2 √2)
```

```
<< SQCS`

numericOff;

H^⊗2 · |0⟩⊗|1⟩ /. expandKron

1/2 (|0⟩⊗|0⟩ - |0⟩⊗|1⟩ + |1⟩⊗|0⟩ - |1⟩⊗|1⟩)

H^⊗5 · H^⊗5

I₅
```

Fig. 6. The Walsh-Hadamard operator in action

The Walsh-Hadamard Operator. Let $\mathrm{H}^{\otimes n}$ denote the Walsh-Hadamard operator on n qubits. It is defined by

$$\mathrm{H}^{\otimes n} |i\rangle = \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} (-1)^{i \cdot j} |j\rangle \ , \tag{1}$$

for each basis state $|i\rangle$ of the Hilbert space $\mathcal{H} = \mathcal{H}_2{}^{\otimes n}$ and where $i \cdot j$ denotes the bitwise inner product of i and j modulo 2.

Using (1) directly in *Mathematica* would require exponentially increasing memory resources. This can be avoided by observing that, given $|\psi\rangle = \otimes_{i=0}^{n-1} |\psi_i\rangle$, where each $|\psi_i\rangle \in \mathcal{H}_2$, the state $\mathrm{H}^{\otimes n} |\psi\rangle$ can be computed in linear time by $\mathrm{H}^{\otimes n} |\psi\rangle = \bigotimes_{i=0}^{n-1} \mathrm{H} |\psi_i\rangle$.

Some properties of the Walsh-Hadamard operator are displayed in fig. 6. Its implementation is based on the general concept of tensor powers of quantum objects, which is fully supported in SQCS. Note that the internal evaluation is a purely symbolic process, hence taking no significant memory resources.

The Outer Product. Consider the Hilbert space $\mathcal{H} = \mathcal{H}_d{}^{\otimes n}$ and two basis states $|i\rangle , |j\rangle \in \mathcal{H}$. The outer product operator $|i\rangle \langle j|$ is defined by the action on each basis state $|k\rangle$, $k = 0, \ldots , d^n - 1$, by $(|i\rangle \langle j|) |k\rangle = \langle j|k\rangle |i\rangle$.

```
<< SQCS`

(| ψ⟩ · ⟨ϕ |) · | η⟩

⟨ϕ | η⟩ | ψ⟩

(| 0⟩ · ⟨1 |)^⊗4 · (| 0⟩⊗| 0⟩⊗| 1⟩⊗| 1⟩)

0

(| 0⟩ · ⟨1 |)^⊗4 · (| 1⟩⊗| 1⟩⊗| 1⟩⊗| 1⟩)

| 0⟩⊗| 0⟩⊗| 0⟩⊗| 0⟩
```

```
<< SQCS`

(Sum[| i4⟩ · ⟨i4 |, {i, 0, 3}]) · | 04⟩ //. expandLinearOutside

| 04⟩
```

Fig. 7. The outer product operator in action

Given a state $|\psi\rangle = \otimes_{s=0}^{n-1}|\psi_s\rangle$, where $|\psi_s\rangle \in \mathcal{H}_2$, the outer-product action $(|i\rangle\langle j|)|\psi\rangle$ can be computed in time $\mathcal{O}(n)$ by first computing radix-2 decompositions of the integers i and j, i.e., $|i\rangle = \otimes_{s=0}^{n-1}|i_s\rangle$, $\langle j| = \otimes_{s=0}^{n-1}\langle j_s|$, and then use the identity $(|i\rangle\langle j|)|\psi\rangle = \bigotimes_{s=0}^{n-1}(|i_s\rangle\langle j_s|)|\psi_s\rangle$.

An important property involving the outer product operator is the completeness relation $\sum_{i=0}^{d^n-1}|i\rangle\langle i| = \mathrm{I}_n$. Figure 7 exhibits several properties of this operator.

3 Simulation of Grover's Algorithm in SQCS

Consider the problem of searching through an unstructured database. Let $N = 2^n$ be the number of records in a database labelled by the integers $0, 1, \ldots, N-1$. Let x_0 be the label of the only record in the database to be searched for.

Any classical algorithm solving this problem must take, on average, $\mathcal{O}(N)$ steps to find x_0. However, Lov Grover [4] devised a quantum algorithm that finds the unknown label, with probability not less than $1-\frac{1}{N}$, in just $\mathcal{O}\left(\sqrt{N}\right)$ steps.

Grover's algorithm was simulated for databases sizes varying from 2^2 to 2^{32}. All the simulations were performed on a Pentium IV, 3.0 GHz processor, 1 GB of RAM, personal computer.

In all the simulations the final measurement step was ignored, in order to overcome the problem of generating a probability distribution over an exponential number of values and then sampling from it (although an efficient implementation of the measurement operator is possible due to the simple structure of the representation of the final state in Grover's algorithm).

Next we briefly review Grover's algorithm. For a more detailed analysis and generalizations see for example [5].

3.1 A Glimpse of Grover's Algorithm

We are given a blackbox function, which we can query as many times as we like, defined by

$$f(x) = \begin{cases} 1 \text{ if } x = x_0 \\ 0 \text{ otherwise} \end{cases} . \tag{2}$$

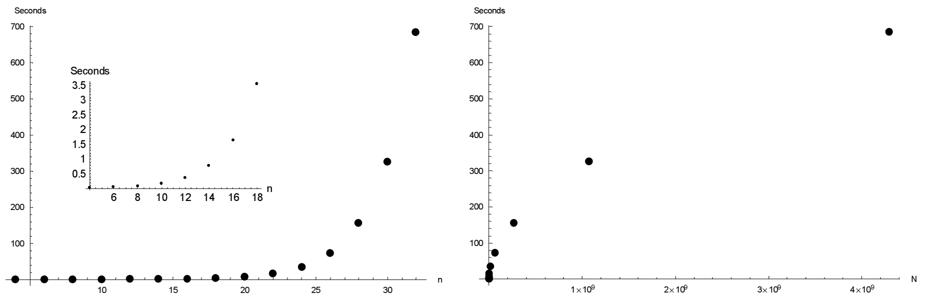

Fig. 8. Simulation times for the quantum database case

This function, for which no knowledge of the internal computational work is assumed, will be called the oracle.

In the quantum perspective, the oracle is given as a blackbox unitary operator U_f defined by $|x\rangle \otimes |y\rangle \;\mapsto\; |x\rangle \otimes |f(x) \oplus y\rangle$, where $\oplus$ denotes the addition modulo 2. U_f is equivalent to the unitary operator $\mathrm{I}_{|x_0\rangle} = \mathrm{I} - 2\,|x_0\rangle\,\langle x_0|$.

Grover's algorithm is based on the operator $\mathrm{Q} = -\,\mathrm{H}^{\otimes n}\;\mathrm{I}_{|0\rangle}\;\mathrm{H}^{\otimes n}\;\mathrm{I}_{|x_0\rangle}$ which performs a rotation of the state of the system towards the desired state $|x_0\rangle$.

Algorithm[Grover].

1. Initialize the state of the system to $|\psi_0\rangle = |0\rangle^{\otimes n}$.
2. Apply the Walsh-Hadamard operator, $|\psi_1\rangle = \mathrm{H}^{\otimes n}\,|\psi_0\rangle$.
3. For $i = 1$ to $k \approx \lfloor \frac{\pi}{4}\sqrt{N} \rfloor$ do $|\psi_{i+1}\rangle = \mathrm{Q}\,|\psi_i\rangle$.
4. Measure $|\psi_{k+1}\rangle$ with respect to the basis $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$.

3.2 Simulations for a Quantum Database

In this case it is assumed that the oracle is given by the operator $\mathrm{I}_{|x_0\rangle}$. The index x_0 is randomly picked up from $\{0, \ldots, 2^n - 1\}$ and no database is in fact generated. When the goal is to test the behaviour and correctness of Grover's algorithm this layout seems adequate.

Simulation Results. Grover's algorithm was simulated over instances of size $n \in \{2, \ldots, 32\}$ qubits (searching on a database with $k = 2^n$ elements). We were able to simulate Grover's algorithm for $n = 30$, using a common PC, in about 5 minutes. As expected, the running times display an exponential growth when taken as a function of the number of qubits, whereas when taken as a function of the database size they follow the square-root curve, as depicted in fig. 8.

3.3 Simulations for a Classical Database

In this case a classical database is first generated and the oracle operator defined by $\mathrm{Oracle} \cdot |x\rangle = (-1)^{f(x)}\,|x\rangle$ where $f(x)$ is the classical oracle call function (2). When applying Grover's operator, we had to fully expand some intermediary states since no useful tensor product decomposition of the oracle operator was found.

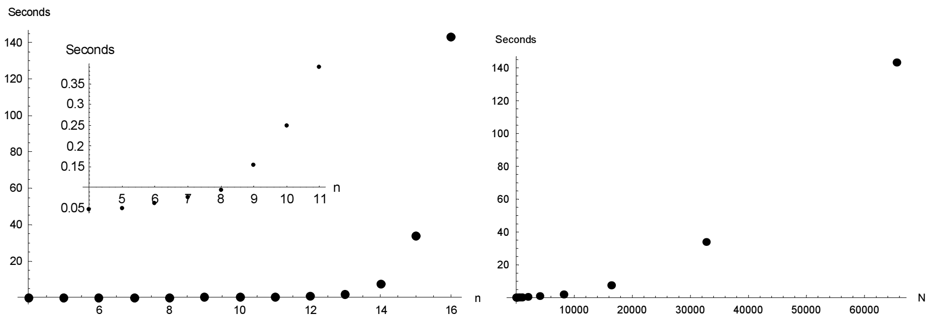

Fig. 9. Simulation times for the classical database case

Simulation Results. Grover's algorithm was simulated over instances of size $n \in \{4, 5, \ldots, 16\}$ qubits. Figure 9 displays the simulation running times as a function of the number of qubits, as well as a function of the database size. Both graphics follow an exponential curve, explained by the inherent inefficiency of the oracle calls in the algorithm.

4 Conclusions

Mathematica can provide a powerful environment for the implementation of a quantum computer simulator, as long as we are able to keep, as much as possible, the implementation at the symbolic level.

A new symbolic approach for the simulation of quantum computations has been proposed. sqcs is an ongoing project still lacking many features. Measuring operators, an improved quantum register address manager and simulations of more quantum algorithms are currently being developed.

References

1. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambrige University Press (2000)
2. Kitaev, A.Y., Shen, A., Vyalyi, M.: Classical and quantum computation. Volume 47 of Graduate Studies in Mathematics. American Mathematical Society (2002)
3. Wolfram, S.: TheMathematica Book, Fifth Edition. Wolfram Media, Inc. (2003)
4. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proc. 28th Annual ACM Symposium on the Theory of Computing. (1996) 212–219
5. Biham, E., Biham, O., Biron, D., Grassl, M., Lidar, D.A.: Grover's quantum search algorithm for an arbitrary initial amplitude distribution. Physical Review A **60** (1999) 27–42