Model Checking Reachability Properties for Quantum Markov Chains

Jens Chr. Godskesen
IT University of Copenhagen

Abstract

We propose a discrete time quantum Markov chain (QMC) related to previous proposals but yet with a different semantics. A probability measure is defined similar to as for DTMCs in contrast to e.g. the super operator valued measure in [2]. Our work is based on a simple imperative quantum pseudo programming language and its denotational semantics which naturally leads to a definition of a QMC. As a novelty we demonstrate how reachability events of a QMC expressed in a simple temporal logic like notation may be checked similar to as for DTMCs as transient state probabilities and how probability intervals of such events may be computed by smallest fixed-point solutions to linear equations.

1 Introduction

Quantum computing had its beginnings in 1982 when Feynman [3] pointed out that a quantum mechanical system can be used to perform computations. The prime factoring algorithm by Shor [9] published in 1994 with its exponential speedup compared to the best performing known algorithms running on traditional computers solving the same problem is a notable contribution illustrating the quantum computing potential. Since the publication of Shor's result there has been much research about quantum algorithms and quantum computing.

While we are still waiting for quantum computers with sufficiently many logical qubits to be able to run algorithms like Shor's we have seen development of quantum programming languages. The first real quantum programming language from 2003 is due to Ömer [7]. An influential paper from 2004 by Sellinger [8] proposes a quantum programming language with a denotational semantics and partial density matrices as denotations.

Formal methods for reasoning about quantum programs and algorithms have been widely studied the last two decades. The paper [4] from 2005 is an early example of using the probabilistic model checker PRISM [6] to verify quantum protocols in a classical probabilistic framework. Later variants of Markov chains, *quantum Markov chains*, were tailored towards modeling and verification of quantum systems. A recent book [11] by Ying et al. gives a state of the art

of model checking quantum systems described by quantum Markov chains and their properties in extensions of classical temporal logics.

The contribution of this paper is a denotational semantics of a simple imperative quantum pseudo programming language, an approach corresponding to the one in [10]. The program semantics naturally leads to the definition of the notion of a discrete time quantum Markov chain (QMC). Our notion of a QMC is similar to the super-operator weighted Markov chain defined in [2] by Ying et al., however we define a probability measure like for DTMCs in contrast to the super-operator valued measure proposed by Ying et al. As a novelty we show how verification of reachability events in a temporal logic like notation may be carried out using verification techniques known from DTMCs. Notably we show how intervals for bounded reachability events may be computed as transient state probabilities and how probability intervals for unbounded constrained reachability events may be computed through the least fixed-point of two sets of linear equations on operators on partial density matrices.

2 A pseudo quantum programming language

We define an imperative quantum pseudo programming language with quantum variables on which unitary operators may be applied. The language contains probabilistic branching and a probabilistic loop construction. The language adopts the idea of "classical control and quantum data" presented in [8].

Assume a finite set of qubit quantum variables ranged over by `q`. The syntax of our pseudo programming language is defined by the grammar:

```
P ::= V begin S end
V ::= qubits r; register r; | register r;
r ::= q,r | q
S ::= 0 | U(r) | S;S | if M[q] then S else S | while M[q] do S end
```

`Prog` is the set of all terms generated by the grammar for `P`, `Var` is the set of all terms generated by the grammar for `V`, and `Stmt` is the set of all terms generated by the grammar for `S`. Brackets may be used to avoid ambiguities.

A program starts with a declaration of quantum variables. Some are local and those occuring after the keywork `register` belong to a register that will contain the program result. Intuitively the meaning of a term in `Stmt` is as follows. `0` is the inactive code. `U(r)` means that a unitary operator `U` is applied to the qubits referred to by the variables `r`. Program parts may be put together in sequence `S;S`. The `if` construction does a mesurement `M` on a single qubit `q`. If the outcome is the state 1 (corresponding to true) the `then` branch is pursued, otherwise the statement affiliated with `else` is carried out. The loop construction measures the state of a single qubit variable, if the outcome is state 1 then the loop continues otherwise it terminates. [1]

[1] Notice that qubits cannot be copied, so the syntax supports the non-cloning property of quantum physics.

```
begin
qubits c;
register p, q;
X(c);
while M[c]
do H(c);
   if M[c] then A+(p,q) else A-(p,q);
   H(c)
end
end
```

Figure 1: The program P0.

Suppose a random walk program P0 over three qubit variables c, p and q as defined in Figure 1. Intuitively c acts as a coin, H is the Hadamard operation on qubits, and X swap qubit states. The two variables p and q constitute a register of two qubits representing the states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ corresponding to the binary representation of the numbers 0, 1, 2 and 3. The operator A+ adds 1 modulo 4 to a state $|i\rangle$, and A- subtracts 1. We let the program be prepared such that all qubits are initialized to $|0\rangle$. Measuring the coin we enter the loop whenever the coin is in state $|1\rangle$, hence the program is probabilistic. Note that the loop will for sure be entered once. The first step in the loop is to apply the Hadamard operation on the coin. Measuring the coin again we may then with equal probability either add or subtract 1 modulo 4 to the register of p and q. The last step in the loop is to apply the Hadamard operation on the coin. The loop will then be entered again with probability a half. What will be the result of running the program? What is the likelihood of the result being an even register value or an odd register value? Will we with equal probability observe an even or odd value? The answer to the latter question is, as we shall see, no.

3 Preliminaries

Let $\mathbb{C}$ be the set of complex numbers. Suppose $n > 0$ qubit *unit vectors* $|\psi_i\rangle \in \mathbb{C}^2$. The qubits constitute a unit vector $|\psi\rangle = |\psi_0\rangle \otimes \cdots \otimes |\psi_{n-1}\rangle$ in $\mathbb{H}$, a finite dimensional Hilbert vector space $\mathbb{C}^{2^n}$ with inner and outer product $\langle\psi|\psi'\rangle \in \mathbb{C}$ and $|\psi\rangle\langle\psi| \in \mathbb{C}^{2^n \times 2^n}$ respectively, and *computational basis* $|0\rangle, \ldots, |2^n - 1\rangle$.

Consider the set of square matrix operators on vectors in $\mathbb{H}$. $\mathcal{H}$ is the set of *Hermetian* matrices, i.e. $M \in \mathcal{H}$ if $M = M^\dagger$ where $M^\dagger$ is the conjugate transpose of M. M is *normal* if $M \cdot M^\dagger = M^\dagger \cdot M$, hence all $M \in \mathcal{H}$ are normal. $\mathcal{P} \subseteq \mathcal{H}$ is the set of *positive* matrices, i.e. $M \in \mathcal{P}$ if $\langle\psi| M |\psi\rangle \geq 0$ for all $|\psi\rangle$. M is *positive definite* if $\langle\psi| M |\psi\rangle > 0$ for all non-zero $|\psi\rangle$.

Let $\mathcal{D} \subseteq \mathcal{P}$ be the set of *partial density matrices*, i.e. $\rho \in \mathcal{D}$ is Hermetian, positive, and $Tr(\rho) \leq 1$. ρ is a density matrices if $Tr(\rho) = 1$. Sometimes we write $\mathcal{D}_n$ to signify $\mathcal{D} \subseteq \mathbb{C}^{2^n \times 2^n}$, i.e. $\rho \in \mathcal{D}_n$ is a partial density matrix over n

qubits. Partial density matrices may be composed using tensor products such that whenever $\rho \in \mathcal{D}_n$ and $\rho' \in \mathcal{D}_m$ then $\rho \otimes \rho' \in \mathcal{D}_{n+m}$. A partial density matrix may be decomposed through a *partial trace mapping* $Tr_{\mathcal{D}_n} : \mathcal{D}_{n+m} \to \mathcal{D}_m$ defined by:

$$Tr_{\mathcal{D}_n}(\rho) = \sum_i (\langle i| \otimes I) \cdot \rho \cdot (|i\rangle \otimes I)$$

where I is the identity matrix on $\mathcal{D}_m$ and $\{|i\rangle\}$ is the computational basis for $\mathcal{D}_n$. It follows that $Tr(\rho) = Tr(Tr_{\mathcal{D}_n}(\rho))$. For instance, whenever $\rho \in \mathcal{D}_n$ and $\rho' \in \mathcal{D}_m$ then

$$\begin{array}{rcl} Tr_{\mathcal{D}_n}(\rho \otimes \rho') & = & \sum_i (\langle i| \otimes I) \cdot \rho \otimes \rho' \cdot (|i\rangle \otimes I) \\ & = & \sum_i (\langle i| \cdot \rho \cdot |i\rangle) \otimes I \cdot \rho' \cdot I \\ & = & Tr(\rho) \cdot \rho' \end{array}$$

$P \in \mathcal{H}$ is a *projection* if $P \cdot P = P$. A finite sequence $\{P_i\}$ of projections is a *projective measurement* if $P_i \cdot P_j = 0$ when $i \neq j$ and $\sum P_i = I$. Letting by convenience $M/p = 0$ when $p = 0$ the outcome of a projective measurement $\{P_i\}$ applied on a partial density matrix ρ is a finite sequence, an *ensemble*, of partial density matrices

$$\{P_i\}(\rho) = \{(p_j, P_j \cdot \rho \cdot {P_j}^\dagger / p_j) \mid p_j = Tr(P_j \cdot \rho), P_j \in \{P_i\}\}$$

where $Tr(\rho) = \sum p_j$. That is, the outcome of the projective measurement is one of $P_i \cdot \rho \cdot {P_i}^\dagger / Tr(P_i \cdot \rho) \in \mathcal{D}$ with probability $Tr(P_i \cdot \rho)$.

Define the (Löwner) partial order $\sqsubseteq$ on $\mathcal{D}$ by $\rho \sqsubseteq \rho'$ if $\rho' - \rho \in \mathcal{P}$. $(\mathcal{D}, \sqsubseteq)$ is a complete partial order (CPO).

Let $(\mathcal{A}, \sqsubseteq)$ be a partial order. Define the *lifted partial order* $(\mathcal{A} \to \mathcal{A}, \sqsubseteq)$ by letting $\mathcal{F} \sqsubseteq \mathcal{G}$ whenever $\mathcal{F}(a) \sqsubseteq \mathcal{G}(a)$ for all a, and for any increasing sequence $\{\mathcal{F}_i\}$ the function $\cup \mathcal{F}_i$ is defined by $(\cup \mathcal{F}_i)(a) = \cup(\mathcal{F}_i(a))$ where $\cup(\mathcal{F}_i(a))$ is the least upper bound of $\{\mathcal{F}_i(a)\}$. If $(\mathcal{A}, \sqsubseteq)$ is complete $(\mathcal{A} \to \mathcal{A}, \sqsubseteq)$ is complete.

For a CPO $(\mathcal{A}, \sqsubseteq)$ $\mathcal{F} : \mathcal{A} \to \mathcal{A}$ is *monotonic* if $\mathcal{F}(a) \sqsubseteq \mathcal{F}(a')$ whenever $a \sqsubseteq a'$. $\mathcal{F}$ is *continuous* if $\mathcal{F}(\cup a_i) = \cup \mathcal{F}(a_i)$ for all increasing sequences $\{a_i\}$.

4 Denotational semantics

In this section we give a denotational semantics for our imperative quantum pseudo programming language. Others, e.g. [10], have proposed similar denotational semantics for quantum programming languages.

Suppose a program `P` with m local variables and $n > 0$ register variables. The denotational semantics for `P` is defined by a function $[\![\ ]\!] : \texttt{Prog} \to \mathcal{D}_n$ where

$$[\![\texttt{V begin S end}]\!] = Tr_{\mathcal{D}_m}([\![\texttt{S}]\!]([\![\texttt{V}]\!]))$$

with $[\![\ ]\!] :$ `Var` $\to \mathcal{D}_{m+n}$ defined by

$$[\![\texttt{qubits r; register s;}]\!] = [\![\texttt{r}]\!] \otimes [\![\texttt{s}]\!]$$
$$[\![\texttt{register r;}]\!] = [\![\texttt{r}]\!]$$
$$[\![\texttt{q, r}]\!] = |0\rangle\langle 0| \otimes [\![\texttt{r}]\!]$$
$$[\![\texttt{q}]\!] = |0\rangle\langle 0|$$

initializing each qubit variable to $|0\rangle$. Note how the part involving local variables is decomposed from the resulting semantics of $[\![\texttt{P}]\!]$.

The semantics of statements is defined by the function

$$[\![\ ]\!](\ ) : \texttt{Stmt} \times \mathcal{D}_{m+n} \to \mathcal{D}_{m+n}$$

For each unitary operator `U(r)` we let $U_{\texttt{U(r)}} \in \mathcal{M}$ be the corresponding unitary matrix (applied on all qubits in the program). A matrix U is unitary if it is normal and $U \cdot U^\dagger = I$. For the inactive code, the unitary operator, and statement composition we have

$$\begin{aligned} [\![\texttt{0}]\!](\rho) &= \rho \\ [\![\texttt{U(r)}]\!](\rho) &= U_{\texttt{U(r)}} \cdot \rho \cdot {U_{\texttt{U(r)}}}^\dagger \\ [\![\texttt{S;T}]\!](\rho) &= [\![\texttt{T}]\!]([\![\texttt{S}]\!](\rho)) \end{aligned}$$

Since $U_{\texttt{U(r)}}$ is a unitary operator $U_{\texttt{U(r)}} \cdot \rho \cdot {U_{\texttt{U(r)}}}^\dagger$ is a partial density matrix if ρ is, and if both $[\![\texttt{T}]\!]()$ and $[\![\texttt{S}]\!]()$ preserves partial density matrices then by induction also $[\![\texttt{S;T}]\!]()$ does.

The two 2 times 2 matrices $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$ are qubit projections into their computational basis and together they constitute a projective measurement $M = \{P_0, P_1\}$. We may be interested in a measurement of a single qubit that is combined with other qubits. Let $j \in \{0, \ldots, m+n-1\}$ be the index for the variable `q` for the qubit we want to measure. Define $M_i = \{P_0, P_1\}$ if $i = j$ and $M_i = \{I\}$ otherwise where I is the 2 times 2 identity matrix, and let

$$M[j] = \{Q_0 \otimes \cdots \otimes Q_{m+n-1} |\ Q_i \in M_i\}$$

Writing $\hat{P}_i$ for $Q_0 \otimes \cdots \otimes P_i \otimes \cdots \otimes Q_{m+n-1}$ then $M[j] = \{\hat{P}_0, \hat{P}_1\}$ is a projective measurement in the computational basis of $\mathcal{C}^{2^{m+n}}$.

Given a pseudo program with $m+n$ qubit variables, let $j \in \{0, \ldots, m+n-1\}$ be the index of the variable `q` in `M[q]`. Letting

$$M[j](\rho) = \{(p_i, \rho_i) \mid p_i = Tr(\hat{P}_i\rho),\ \rho_i = \hat{P}_i\rho\hat{P}_i^\dagger/p_i,\ \hat{P}_i \in M[j]\}$$

the semantics of the `if` statement is as a partial density matrix being a linear combination of partial density matrices

$$[\![\texttt{if M[q] then S else T}]\!](\rho) = p_0[\![\texttt{T}]\!](\rho_0) + p_1[\![\texttt{S}]\!](\rho_1)$$

assuming by induction $[\![\texttt{S}]\!]()$ and $[\![\texttt{T}]\!]()$ preserves partial density matrices. Note that we let the projection $\hat{P}_0$ represent "false" and $\hat{P}_1$ represents "true"

For the semantics of the `while` construct let `q` in `M[q]` be the qubit indexed by j. This gives like above a projective measurement $M[j] = \{\hat{P}_0, \hat{P}_1\}$. Assuming by induction $[\![\mathtt{S}]\!]()$ preserves partial density matrices, define two endo-functions $\mathcal{F}_0$ and $\mathcal{F}_1$ on $\mathcal{D}_{m+n}$ by

$$\mathcal{F}_0(\rho) = \hat{P}_0 \rho \hat{P}_0^\dagger \qquad \mathcal{F}_1(\rho) = [\![\mathtt{S}]\!](\hat{P}_1 \rho \hat{P}_1^\dagger)$$

then letting $\mathcal{F}^0 = I$

$$[\![\texttt{while M[q] do S end}]\!](\rho) = \sum_{i=0}^{\infty} \mathcal{F}_0({\mathcal{F}_1}^i(\rho))$$

I.e. the semantics of the `while` construction is the least fixed-point for the continuous higher order endo-function $\mathcal{F}_{\texttt{while}}$ on $\mathcal{D}_{m+n} \to \mathcal{D}_{m+n}$ where

$$\mathcal{F}_{\texttt{while}}(\mathcal{F}) \quad = \quad \mathcal{F}_0 + \mathcal{F}(\mathcal{F}_1)$$

The least fixed-point is due to Knaster-Tarski the least upper bound of the increasing sequence: $0 \sqsubseteq \mathcal{F}^1_{\texttt{while}}(0) \sqsubseteq \mathcal{F}^2_{\texttt{while}}(0) \sqsubseteq \ldots \sqsubseteq \mathcal{F}^j_{\texttt{while}}(0) \ldots$

4.1 Termination

Programs may not terminate and loop indefinitely. As an example, consider

```
begin
register q;
X(q);
while M[q] do 0 end
end
```

Cleary this is an infinite loop, it terminates with probability 0. Each element in the infinite sum $\sum_{j=0}^{\infty} \mathcal{F}_0(\mathcal{F}_1^j(|1\rangle\langle 1|))$ is a zero matrix so the denotational semantics is the zero matrix with trace equal to 0.

$Tr([\![\mathtt{S}]\!](\rho))$ is the probability `S` *terminates* when started in state ρ. One may show that $Tr([\![\mathtt{S}]\!](\rho)) \leq Tr(\rho)$.

4.2 Example

Recall the probabilistic example program `P0` in Figure 1 where the tossing of a coin, the qubit `c`, influences a random walk on the values of the register consisting of two qubits `p`, and `q`.

The program starts in a state where all the qubits are initialized to $|0\rangle$. Hence the program initial state is $|000\rangle\langle 000|$. The first step of the program is to swap the state of the qubit for the variable `c` so the density matrix then becomes $\rho = |100\rangle\langle 100|$.

By convenience we let a partial density matrix be written as a sum of elements $\sum p_i \rho_i$ where p_i is a probability and ρ_i is a partial density matrix

$\rho_i = |0\rangle \otimes |i\rangle \cdot \langle 0| \otimes \langle i|$ where the label i ranges over the binary numbers 00, 01, 10 and 11.

The local variable `c` is indexed by 0 so let $i = 0$ in the definition of the functions $\mathcal{F}_0$ and $\mathcal{F}_1$ in the denotational semantics. Then the semantics of the loop, $\mathcal{F}_{\texttt{while}}(\rho)$, when the loop as input is given the density matrix ρ is

$$\begin{aligned}\mathcal{F}_{\texttt{while}}(\rho) &= 0 + \mathcal{F}_0(\rho) + \mathcal{F}_0(\mathcal{F}_1(\rho)) + \mathcal{F}_0(\mathcal{F}_1^2(\rho)) + \mathcal{F}_0(\mathcal{F}_1^3(\rho)) + \cdots \\ &= \textstyle\sum_{n=1}^{\infty} \frac{1}{2^{2n+1}}\rho_{00} + \frac{1}{2^{2n}}\rho_{01} + \frac{1}{2^{2n+1}}\rho_{10} + \frac{1}{2^{2n}}\rho_{11} \\ &= \tfrac{1}{6}\rho_{00} + \tfrac{1}{3}\rho_{01} + \tfrac{1}{6}\rho_{10} + \tfrac{1}{3}\rho_{11}\end{aligned}$$

Decomposing the part of $\mathcal{F}_{\texttt{while}}(\rho)$ relating to the subsystem being the local variable `c` the semantics of `P0` is $Tr_{\mathcal{D}_1}(\mathcal{F}_{\texttt{while}}(\rho))$, i.e.

$$[\![\text{P0}]\!] = \frac{1}{6}|00\rangle\langle 00| + \frac{1}{3}|01\rangle\langle 01| + \frac{1}{6}|10\rangle\langle 10| + \frac{1}{3}|11\rangle\langle 11|$$

Hence the probability of observing an odd versus an even register value is 2/3 and 1/3 respectively.

It seems that variants of `P0` where the value of the coin is manipulated differently before and inside the loop all will produce an outcome where the register values are not evenly distributed. So how to define a variant of `P0` where the result is an even distribution of the register values? We could e.g. make use of *superposition*. Let the program `Q0` be `P0` but with `HH(p,q)` added just before entering the loop. The meaning of `HH` is $I \otimes H \otimes H$ which brings the register in a balanced superposition possessing all the potential four register values simultaneously with equal probabilities. Consequently, when the register is incremented (or decremented) all the four values will simultaneously be incremented (decremented) leaving the register unchanged. Hence the semantics of `Q0` is the balanced superposition

$$[\![\texttt{Q0}]\!] = \frac{1}{4}|00\rangle\langle 00| + \frac{1}{4}|01\rangle\langle 01| + \frac{1}{4}|10\rangle\langle 10| + \frac{1}{4}|11\rangle\langle 11|$$

5 Quantum Markov Chains

In [2] the authors define a super-operator weighted Markov chain based on a countable set of states and a transition function where pair of states are mapped to a so-called super-operator over a Hilbert space. Below we adopt a slightly different approach and define a variant of a quantum Markov chain inspired by [5] as a pair $(\mathbb{M}, \Delta)$ where $\mathbb{M}$ is a square matrix with super-operator entries and Δ is a vector of partial density matrices on which $\mathbb{M}$ may be applied. In contrast to the super-operator valued measure in [2] we define instead a probability measure similar to as for DTMCs.

5.1 Example

Consider the program `Q1` and its associated graph in Figure 2. We let labels on the graph represent partial density matrix operators $\mathcal{F}_U(\rho) = U \cdot \rho \cdot U^\dagger$ when

```
begin
register q;
while M[q] do H(q) end
end
```

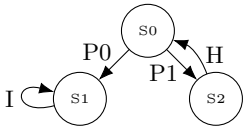

Figure 2: The code for `Q1` and its graph representation.

U is unitary and $\mathcal{F}_P(\rho) = P \cdot \rho \cdot P^\dagger$ where P is a projection. $\{P_0, P_1\}$ is the projective measurement on the qubit referred to by the variable `q`.

Intuitively the program starts in the initial state `S0` in which a projective measurement is made on its variable `q`. If the outcome is 0 the state `S1` is entered and never left, otherwise the program enters state `S2` and returns to the initial state after a Hadamard operation on `q`. `Q1` can be represented by the 3 times 3 matrix $\mathbb{M}$ with density matrix operators where the indices $0, 1, 2$ corresponds to the program states `S0`, `S1`, and `S2` respectively.

$$\mathbb{M} = \begin{pmatrix} 0 & 0 & \mathcal{F}_H \\ \mathcal{F}_{P_0} & I & 0 \\ \mathcal{F}_{P_1} & 0 & 0 \end{pmatrix}$$

$\mathbb{M}_{i,j}$, corresponds to the operator applied when going from program state j to i. So, the entries in column j are the outgoing operations from program state j and the entries in row i are the incoming operations to program state i.

Assume an initial system state vector $\Delta = (\rho_0, 0, 0)$, i.e. the state `S0` has the initial value $\rho_0 = |0\rangle\langle 0|$. We call the tupple $\mathcal{M}_{\mathtt{Q1}} = (\mathbb{M}, \Delta)$ for a quantum Markov chain. Applying $\mathbb{M}$ on Δ yields the system state vector $\mathbb{M}\Delta = (0, \mathcal{F}_{P_0}(\rho_0), 0) = (0, \rho_0, 0)$ and applying $\mathbb{M}$ once again gives $\mathbb{M}(\mathbb{M}\Delta) = \mathbb{M}\Delta$, hence the evolution of `Q1` is the periodic system state vector path

$$\Delta(\mathbb{M}\Delta)(\mathbb{M}\Delta)\ldots(\mathbb{M}\Delta)$$

Since `S1` is the terminal state in `Q1` we may represent the semantics of `Q1` by

$$[\![\mathtt{Q1}]\!] = \lim_{n\to\infty} (\mathbb{M}^n\Delta)[1]$$

where $(\mathbb{M}^n\Delta)[1]$ is entry 1 in $\mathbb{M}^n\Delta$.

Because $\mathcal{F}_{P_1}(\Delta[0]) = 0$ there is only one path of the system, i.e.

$$\pi_{\mathtt{Q1}} = (0, \rho_0)(1, \rho_0)(1, \rho_0)\ldots$$

relative to the sequence of operators $\mathbb{M}_{1,0}, \mathbb{M}_{1,1}, \mathbb{M}_{1,1}, \ldots$

Observe that we are operating with states at two levels, one kind of states are the program states `S0`, `S1`, and `S2`, the other kind of states are the state vectors of the associated Hilbert space represented as density matrices.

Suppose instead we have an initial system state vector $\Delta' = (\rho', 0, 0)$ where $\rho' = |+\rangle\langle+|$. Then letting $\rho_1 = |1\rangle\langle 1|$ we get:

$$\begin{aligned}
\mathbb{M}^1\Delta' &= (0, \rho_0/2, \rho_1/2) \\
\mathbb{M}^2\Delta' &= (\rho'/2, \rho_0/2, 0) \\
\mathbb{M}^3\Delta' &= (0, 3\rho_0/4, \rho_1/4) \\
\mathbb{M}^4\Delta' &= (\rho'/4, 3\rho_0/4, 0)
\end{aligned}$$

In general, if $\Delta_0^n = (0, (2^n-1)\rho_0/2^n, \rho_1/2^n)$ and $\Delta_1^n = (\rho'/2^n, (2^n-1)\rho_0/2^n, 0)$ we have the non-periodic evolution of system state vectors

$$\Delta'\Delta_0^1\Delta_1^1\Delta_0^2\Delta_1^2 \ldots \Delta_0^i\Delta_1^i \ldots$$

In this case the system has infinitely many periodic paths:

$$\begin{aligned}
\pi_0 &= (0, \rho')(1, \rho_0/2)(1, \rho_0/2)\ldots \\
\pi_1 &= (0, \rho')(2, \rho_1/2)(0, \rho'/2)(1, \rho_0/4)(1, \rho_0/4)\ldots \\
\pi_2 &= (0, \rho')(2, \rho_1/2)(0, \rho'/2)(2, \rho_1/4)(0, \rho'/4)(1, \rho_0/8)(1, \rho_0/8)\ldots \\
&\vdots
\end{aligned}$$

5.2 Quantum Markov Chains

We next define the notion of a discrete time quantum Markov Chain (QMC).

Suppose $\mathbb{H}$ with partial density matrices $\mathcal{D}$. Let $\mathbb{E}$ be the set of endo-functions on $\mathcal{D}$ and define $(\mathcal{E}+\mathcal{F})(\rho) = \mathcal{E}(\rho) + \mathcal{F}(\rho)$ and $(\mathcal{E}\circ\mathcal{F})(\rho) = \mathcal{E}(\mathcal{F}(\rho))$ for all $\mathcal{E}, \mathcal{F} \in \mathbb{E}$. $(\mathbb{E}, +, \circ)$ forms a semiring with monoids $(\mathbb{E}, +, 0)$ and $(\mathbb{E}, \circ, I)$. Define the preorder $\leq$ by $\mathcal{E} \leq \mathcal{F}$ if $Tr(\mathcal{E}(\rho)) \leq Tr(\mathcal{F}(\rho))$ for all $\rho \in \mathcal{D}$. We write $\mathcal{E} \simeq \mathcal{F}$ if $\mathcal{E} \leq \mathcal{F}$ and $\mathcal{F} \leq \mathcal{E}$ and $\mathcal{E} < \mathcal{F}$ if $\mathcal{E} \leq \mathcal{F}$ but $\mathcal{E} \not\simeq \mathcal{F}$. Obviously $0 \leq \mathcal{F}$ for all $\mathcal{F}$. The preorder $\leq$ contains the partial order $\sqsubseteq$.

We let $\mathbb{O}$ be the set of endo-functions, *operators*, on $\mathcal{D}$ such that $\mathcal{F} \leq I$ for all $\mathcal{F} \in \mathbb{O}$. As an example, the two functions $\mathcal{F}_U$ and $\mathcal{F}_P$ belong to $\mathbb{O}$ for all unitaries U and projections P. Also $\mathcal{F}_P \leq \mathcal{F}_U \simeq I$ for all P and U. Obviously for any two $\mathcal{E}, \mathcal{F} \in \mathbb{O}$ the composition $\mathcal{E}\circ\mathcal{F} \in \mathbb{O}$.

We let a vector $\Delta = (\rho_0, \ldots, \rho_{n-1})$ have trace $Tr(\Delta) = \sum_i Tr(\rho_i)$ and call it a *system state vector* if $Tr(\Delta) = 1$. When $\Delta = (\rho_0, \ldots, \rho_{n-1})$ we let $\Delta[i] = \rho_i$. Intuitively, a system state vector Δ represents a state where the system is in state i with probability $Tr(\Delta[i])$ and $\Delta[i]$ is the state for i in the associated Hilbert space. For $\mathbb{M} \in \mathbb{O}^{n\times n}$ the matrix entry row i and column j is denoted $\mathbb{M}_{i,j}$. $\mathbb{M}$ is an *operator* on $\mathcal{D}^n$ if $\sum_i \mathbb{M}_{i,j} \simeq I$ for all j.

Definition 1 *A QMC over $\mathbb{H}$ is a tupple $\mathcal{M} = (\mathbb{M}, \Delta)$ where $\Delta \in \mathcal{D}^n$ is an initial system state vector and $\mathbb{M}$ is an operator on $\mathcal{D}^n$.*

We sometimes write $\mathcal{M}_\Delta$ for $\mathcal{M} = (\mathbb{M}, \Delta)$.

A QMC is a generalization of finite discrete time Markov chains without atomic propositions since any such Markov chain $\mathcal{M}$ with n states can be encoded as $(\mathbb{M}, \Delta)$ over a one dimensional Hilbert space with $\Delta = (\rho_0, \ldots, \rho_{n-1})$

where $Tr(\rho_i)$ is the probability of initially being in the i'th state of $\mathcal{M}$ and $\mathbb{M}_{i,j}$ is the probability of entering state i from state j.

Given $\mathcal{M} = (\mathbb{M}, \Delta)$ where $\Delta \in \mathcal{D}^n$ a *path* of $\mathcal{M}$ is an infinite sequence of *path states*, i.e. pairs of indices and partial density matrices,

$$\pi = (i_0, \rho_0)(i_1, \rho_1) \dots (i_j, \rho_j) \dots$$

with $i_k \in \{0, \dots, n-1\}$, $Tr(\rho_k) > 0$, $\rho_0 = \Delta[i_0]$, and $\rho_{k+1} = \mathbb{M}_{i_{k+1}, i_k}(\rho_k)$. $Path_{\mathcal{M}}$ is the set of all paths of $\mathcal{M}$. $\pi[i]$ is the i'th element of π. $prefix(\pi)$ is the set of all prefixes of π and $Path^{fin}_{\mathcal{M}} = \{\hat{\pi} \in prefix(\pi) \mid \pi \in Path_{\mathcal{M}}\}$. $|\hat{\pi}|$ is the length of $\hat{\pi}$ and $last(\hat{\pi})$ is the last pair in $\hat{\pi}$ if $|\hat{\pi}| > 0$.

The *evolution* of $\mathcal{M} = (\mathbb{M}, \Delta)$ is an infinite sequence of system state vectors defined as a series of applications of $\mathbb{M}$ on Δ, i.e. the evolution of $\mathcal{M}$ is

$$\Delta(\mathbb{M}\Delta)(\mathbb{M}^2\Delta) \dots (\mathbb{M}^i\Delta) \dots$$

$\mathbb{M}$ is trace preserving because $\mathbb{M}$ is an operator, i.e. $Tr(\mathbb{M}^i\Delta) = 1$ for all i. Also we may by induction in i prove that

Lemma 1 *For all i $\sum_j \{ Tr(\rho_j) \mid \pi \in Path_{\mathcal{M}}, \pi[i] = (j, \rho_j)\} = 1$*

5.3 Probability Space

Let $Path_{\mathcal{M}}$ be the set of possible *outcomes* in the probability space for $\mathcal{M}$. We have to choose a non-empty collection of subsets of $Path_{\mathcal{M}}$ as *events* to which we want to assign probabilities. This set of events must form a σ-field. Like for ordinary Markov chains we let the set of events be induced by cylinder sets.

For all $\hat{\pi} \in Path^{fin}_{\mathcal{M}}$ define the cylinder set $Cyl_{\mathcal{M}}(\hat{\pi})$ by

$$Cyl_{\mathcal{M}}(\hat{\pi}) = \{\pi \in Path_{\mathcal{M}} \mid \hat{\pi} \in prefix(\pi)\}$$

Hence $Cyl_{\mathcal{M}}(\hat{\pi})$ is the set of all paths in $Path_{\mathcal{M}}$ starting with $\hat{\pi}$. Note that if $\hat{\pi}$ is the empty prefix ϵ then $Cyl_{\mathcal{M}}(\hat{\pi}) = Path_{\mathcal{M}}$.

Definition 2 *Let $\Sigma_{\mathcal{M}}$ be the least σ-field that for all $\hat{\pi} \in Path^{fin}_{\mathcal{M}}$ contains $Cyl_{\mathcal{M}}(\hat{\pi})$.*

The probability for a cylinder set is defined by $Pr_{\mathcal{M}}(Cyl_{\mathcal{M}}(\epsilon)) = 1$ and whenever $last(\hat{\pi}) = (i, \rho)$ for some i then

$$Pr_{\mathcal{M}_\Delta}(Cyl_{\mathcal{M}_\Delta}(\hat{\pi})) = Tr(\rho) \tag{1}$$

We let $Pr_{\mathcal{M}}(\emptyset) = 0$. If $\{Cyl_{\mathcal{M}}(\hat{\pi}_i)\}$ is a collection of pairwise disjoint sets we define

$$Pr_{\mathcal{M}}(\cup Cyl_{\mathcal{M}}(\hat{\pi}_i)) = \sum Pr_{\mathcal{M}}(Cyl_{\mathcal{M}}(\hat{\pi}_i))$$

Hence $Pr_{\mathcal{M}}$ is a premeasure on $\Sigma_{\mathcal{M}}$ and from classical probability theory there exists a unique extension of $Pr_{\mathcal{M}}$ to a probability measure on $\Sigma_{\mathcal{M}}$.

For $\mathcal{M}_{\mathtt{Q1}}$ defined in Figure 2 $Path_{\mathcal{M}_{\mathtt{Q1}}} = \{\pi_{\mathtt{Q1}}\}$. Hence $Cyl_{\mathcal{M}_{\mathtt{Q2}}}(\hat{\pi}) = \{\pi_{\mathtt{Q1}}\}$ for any finite path $\hat{\pi}$ of $\mathcal{M}_{\mathtt{Q1}}$. The least σ-field for $\mathcal{M}_{\mathtt{Q1}}$ obviously is $\{\emptyset, \{\pi_{\mathtt{Q1}}\}\}$ and $Pr_{\mathcal{M}_{\mathtt{Q1}}}(\{\pi_{\mathtt{Q1}}\}) = 1$.

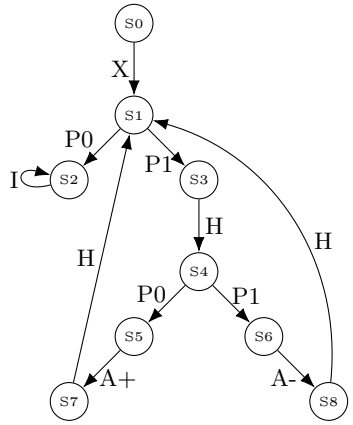

Figure 3: Graph representation for P0.

$$\mathbb{M}_{\texttt{P0}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathcal{F}_X & 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{F}_H & \mathcal{F}_H \\ 0 & \mathcal{F}_{P_0} & \mathcal{F}_I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{F}_{P_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{F}_H & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{F}_{P_0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{F}_{P_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{F}_{A_+} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{F}_{A_-} & 0 & 0 \end{pmatrix}$$

Figure 4: Matrix representation for the graph for P0.

5.4 Example

We may define a QMC $\mathcal{M}_{\texttt{P0}} = (\mathbb{M}_{\texttt{P0}}, \Delta)$ for the program P0 defined in Figure 1 with $\Delta = (\rho_0, 0, 0, 0, 0, 0, 0, 0, 0)$ where $\rho_0 = |000\rangle\langle 000|$ and with the graph for P0 depicted in Figure 3 where $\{P_0, P_1\}$ is the projective measurement on the qubit referred to by the variable c. The matrix $\mathbb{M}_{\texttt{P0}}$ is defined in Figure 4.

Since S2 is the final state of P0 its denotational semantics may be defined as the limit of the evolution of $\mathcal{M}_{\texttt{P0}}$ for the index representing S2 while disregarding the part of the system representing the variable c, i.e.

$$[\![\texttt{P0}]\!] = Tr_{\mathcal{D}_1}(\lim_{n\to\infty}(\mathbb{M}_{\texttt{P0}}{}^n\Delta)[2])$$

Writing ρ_{ij} for $|ij\rangle\langle ij|$ and ρ_{ijk} for $|ijk\rangle\langle ijk|$ the path

$$\hat{\pi}_0 = (0, \rho_0)(1, \rho_{100})(3, \rho_{100})(4, |-\rangle\langle -| \otimes \rho_{00})$$

is the initial finite path for all paths in $Path_{\mathcal{M}_{\texttt{P0}}}$ and $Pr_{\mathcal{M}_{\texttt{P0}}}(Cyl_{\mathcal{M}_{\texttt{P0}}}(\hat{\pi}_0)) = 1$. After $\hat{\pi}_0$ a measurement of the coin is done and the two following paths bring `P0` back to `S1` after having incremented and decremented respectively the register

$$\hat{\pi}_1 = \hat{\pi}_0(5, \tfrac{1}{2}\rho_{000})(7, \tfrac{1}{2}\rho_{001})(1, |+\rangle \langle +| \otimes \tfrac{1}{2}\rho_{01})$$
$$\hat{\pi}_2 = \hat{\pi}_0(6, \tfrac{1}{2}\rho_{100})(8, \tfrac{1}{2}\rho_{111})(1, |-\rangle \langle -| \otimes \tfrac{1}{2}\rho_{11})$$

Notice $Pr_{\mathcal{M}_{\texttt{P0}}}(Cyl_{\mathcal{M}_{\texttt{P0}}}(\hat{\pi}_1)) = Pr_{\mathcal{M}_{\texttt{P0}}}(Cyl_{\mathcal{M}_{\texttt{P0}}}(\hat{\pi}_2)) = 0.5$. $\hat{\pi}_1$ and $\hat{\pi}_2$ each have two continuations

$$\hat{\pi}_3 = \hat{\pi}_1(2, \tfrac{1}{4}\rho_{001})$$
$$\hat{\pi}_4 = \hat{\pi}_2(2, \tfrac{1}{4}\rho_{011})$$
$$\hat{\pi}_5 = \hat{\pi}_1(3, \tfrac{1}{4}\rho_{101})(4, |-\rangle \langle -| \otimes \tfrac{1}{4}\rho_{01})$$
$$\hat{\pi}_6 = \hat{\pi}_2(3, \tfrac{1}{4}\rho_{111})(4, |-\rangle \langle -| \otimes \tfrac{1}{4}\rho_{11})$$

where $Pr_{\mathcal{M}_{\texttt{P0}}}(Cyl_{\mathcal{M}_{\texttt{P0}}}(\hat{\pi}_i)) = 0.25$, $i = 3, 4, 5, 6$. $\hat{\pi}_3$ and $\hat{\pi}_4$ lead to

$$Cyl_{\mathcal{M}_{\texttt{P0}}}(\hat{\pi}_3) = \{\hat{\pi}_1(2, \tfrac{1}{4}\rho_{001})^\omega\}$$
$$Cyl_{\mathcal{M}_{\texttt{P0}}}(\hat{\pi}_4) = \{\hat{\pi}_2(2, \tfrac{1}{4}\rho_{011})^\omega\}$$

where the register is equal to 1 and 3 respectively with equal probability 0.25.

In general for $n \geq 1$ we have paths with odd register value ij

$$\pi_n^{ij} = \hat{\pi}_n^{ij}(2, \frac{1}{2^{2n}}\rho_{0ij})^\omega$$

for some $\hat{\pi}_n^{ij}$ with $Pr_{\mathcal{M}_{\texttt{P0}}}(\{\pi_n^{ij}\}) = \frac{1}{2^{2n}}$, and paths with even register value ij

$$\pi_n^{ij} = \hat{\pi}_n^{ij}(2, \frac{1}{2^{2n+1}}\rho_{0ij})^\omega$$

for some $\hat{\pi}_n^{ij}$ with probability $Pr_{\mathcal{M}_{\texttt{P0}}}(\{\pi_n^{ij}\}) = \frac{1}{2^{2n+1}}$.

6 Reachability Events

How to reason about programs in our programming language? How e.q. to calculate the probability of reaching a set of states in a program execution? For instance, for the program `P0` in Figure 1 we may ask what is the probability of the event "there is a path to `S2` such that the register value is 1".

Suppose $\mathcal{M} = (\mathbb{M}, \Delta)$ over $\mathbb{H}$ with computational basis $\{|0\rangle, \ldots, |2^n - 1\rangle\}$ and where $\mathbb{M}$ is an m square matrix. Using a temporal logic like notation we may write $\Diamond S$ for the event "a state in S will eventually be reached" where S represents a set of path states and is a non-empty subset of

$$\{(i, |j\rangle) \,|\, i \in \{0, \ldots, m-1\}, j \in \{0, \ldots, 2^n - 1\}\}$$

We say (i, ρ) *satisfies* S and write $(i, \rho) \vdash S$ if there exists $(i, |j\rangle) \in S$ such that $Tr(|j\rangle \langle j| \rho) > 0$. We say i satisfies S and write $i \vdash S$ if $(i, \rho) \vdash S$ for some

ρ. We write $\hat{\pi} \vdash S$ if $last(\hat{\pi}) \vdash S$. We define $S_{i,j} = \{(i, |j\rangle)\}$ and let S_i be the set $\cup_j S_{i,j}$. Note that $\{(i, |j\rangle)\}$ may be regarded a qualitative proposition which (i, ρ) satisfies or not. $\cup S_{i,k}$ is a disjunction of all the propositions $S_{i,k}$. Intuitively (i, ρ) satisfies $\cup_k S_{i,k}$ if the system states in $\mathbb{H}$ the partial density matrix ρ represents are contained in the subspace being the span of $\sum |k\rangle$.

For the reachability event $\Diamond S$ we formally define

$$Pr_{\mathcal{M}}(\Diamond S) = \sum_{\hat{\pi} \in Path_{\mathcal{M}}(S)} Pr_{\mathcal{M}}(Cyl_{\mathcal{M}}(\hat{\pi})) \tag{2}$$

with $Path_{\mathcal{M}}(S) = \{\hat{\pi} \in Path^{fin}_{\mathcal{M}} \mid \hat{\pi} \vdash S, \forall \hat{\pi}' < \hat{\pi}. \hat{\pi}' \nvdash S\}$ writing $\hat{\pi}' < \hat{\pi}$ if $\hat{\pi}'$ is a prefix of $\hat{\pi}$ and $\hat{\pi} \neq \hat{\pi}'$. Observe that the cylinder sets in Equation 2 are pairwise disjoint and since $Path(S)$ is countable the event $\Diamond S$ is measurable.

6.1 Transient state probabilities

Given a QMC $\mathcal{M} = (\mathbb{M}, \Delta)$ the system state vector $\Delta_n = \mathbb{M}^n \Delta$ contains the probability $Tr(\Delta_n[i])$ of being in state $(i, \Delta_n[i])$ after n steps of evolution of $\mathcal{M}$. We call such a probability a *transient state probability* for i.

Suppose an event $\Diamond S_i$. Its bounded variant $\Diamond^{\leq n} S_i$ saying that S_i must be reached in at most n evolution steps can be computed as a transient state probability for i by $\mathcal{M}' = (\mathbb{N}, \Delta)$ where $\mathbb{N}$ is as $\mathbb{M}$ except that i must be *absorbing*, that is $\mathbb{N}_{i,i} = I$ and $\mathbb{N}_{j,i} = 0$ for all $j \neq i$. Note that $\mathbb{N}$ indeed is an operator so $\mathcal{M}'$ is well-defined. It is obvious that $Pr_{\mathcal{M}}(\Diamond^{\leq n} S_i) = Pr_{\mathcal{M}'}(\Diamond^{=n} S_i)$ and $Pr_{\mathcal{M}'}(\Diamond^{=n} S_i) = Tr(\Delta'[i])$ where $\Delta' = \mathbb{N}^n \Delta$.

If the event is $\Diamond^{\leq n} S_{i,j}$ we compute Δ' as above and identify and interval by the projection $P = |j\rangle \langle j|$ such that

$$p_0 \leq Pr_{\mathcal{M}}(\Diamond^{\leq n} S_{i,j}) \leq p_1$$

where $p_0 = Tr(\mathcal{F}_P(\Delta'[i]))$ and $p_1 = Tr(\Delta'[i])$ if $Tr(\mathcal{F}_P(\Delta'[i])) > 0$, otherwise $p_1 = 0$. Due to Lemma 1, intuitively p_0 is the sum of the probabilities of P for each finite paths leading to i, and p_1 is the sum of the probabilities for each finite path leading to i if at least one path satisfies P.

More generally if we have $\Diamond^{\leq n} S$ where $S = \cup S_{i,j}$ for a fixed i and several j's we compute p_0 and p_1 as above but where $P = \sum_{(i,|j\rangle) \in S} |j\rangle \langle j|$.

6.2 Constrained Reachability

Reconsider the graph in Figure 3 and suppose we want to reach program state `S2`, but only through certain other program states and requiring only specific register values in `S2`. The event may for instance be "`S2` will eventually be reached such that the register value has only been incremented and is 1". This is a *constrained reachability event* and can be formalized by $T_{\mathtt{S2}} \,\mathbf{U}\, S_{2,1}$ where

$$T_{\mathtt{S2}} \quad = \quad S_0 \cup S_1 \cup S_3 \cup S_4 \cup S_5 \cup S_7$$

We define $T \,\mathbf{U}\, S$ as a generalization of $\Diamond S$ by

$$Pr_{\mathcal{M}}(T \,\mathbf{U}\, S) = \sum_{\hat{\pi} \in Path_{\mathcal{M}}(T \,\mathbf{U}\, S)} Pr_{\mathcal{M}}(Cyl_{\mathcal{M}}(\hat{\pi})) \tag{3}$$

where $Path_{\mathcal{M}}(T \,\mathbf{U}\, S) = \{\hat{\pi} \in Path^{fin}_{\mathcal{M}} \mid \hat{\pi} \vdash S, \forall \epsilon < \hat{\pi}' < \hat{\pi}.\, \hat{\pi}' \vdash T, \hat{\pi}' \nvdash S\}$. Observe that also in this case the cylinder sets in Equation 3 are pairwise disjoint and since $Path(T \,\mathbf{U}\, S)$ is countable the event $T \,\mathbf{U}\, S$ is measurable.

The bounded variant of $T \,\mathbf{U}\, S$ is defined by

$$Pr_{\mathcal{M}}(T \,\mathbf{U}^{\leq n}\, S) = \sum_{\hat{\pi} \in Path_{\mathcal{M}}(T \,\mathbf{U}^{\leq n}\, S)} Pr_{\mathcal{M}}(Cyl_{\mathcal{M}}(\hat{\pi}))$$

where $Path_{\mathcal{M}}(T \,\mathbf{U}^{\leq n}\, S) = \{\hat{\pi} \in Path_{\mathcal{M}}(T \,\mathbf{U}\, S) \mid |\hat{\pi}| \leq n\}$.

6.2.1 Transient state probabilities

As we saw in Section 6.1 the probability of bounded reachability events may be found as transient state probabilities, and the same holds for constrained reachability events on the form $\cup T_i \,\mathbf{U}^{\leq n}\, S$.

Assume a QMC $\mathcal{M} = (\mathbb{M}, \Delta)$. Computing an interval for $Pr_{\mathcal{M}}(\cup T_i \,\mathbf{U}^{\leq n}\, S)$ we modify $\mathbb{M}$ such that $\mathbb{N}$ in $\mathcal{M}' = (\mathbb{N}, \Delta)$ is $\mathbb{M}$ except that all j either satisfying S or not satisfying $\cup T_i$ become absorbing in $\mathbb{N}$. Intuitively in the revised $\mathbb{N}$ one stays with probability 1 in nodes satisfying S and nodes not satisfying $\cup T_i$ cannot be traversed. Obviously $\mathbb{N}$ is then an operator and $\mathcal{M}'$ is well-defined.

If $S = \cup_k S_{j,k}$, hence as in Section 6.1 we may first compute $\mathbb{N}^n\Delta$ and then make projections.

6.2.2 Example

Suppose we are interested in calculating the probabilities of pure constrained reachability events and not their bounded versions. One can imagine that such probabilities may be calculated as transient state probabilities over the limit of evolutions of QMCs. However, such computations may be infinite and rather cumbersome in case of large matrices. Instead, we may turn to equation solving as is common for finite DTMC's, see e.g. [1]. We hence devise a two-phase algorithm where first a graph analysis is performed to compute the subset of nodes that can reach the goal nodes in one or more steps. In the second phase we solve a linear equation of operators on partial density matrices.

Let us look at an example. How e.g. to calculate the probability of the event $T_{\mathtt{S2}} \,\mathbf{U}\, S_{2,1}$ for `P0` defined in Figure 1 with its graph representation in Figure 3? The first part of the algorithm identifies that nodes 2, 6, and 8 cannot reach the goal node 2 in one or more steps. Hence we define a matrix $A_{\mathtt{S2}}$ leaving out the

rows and columns for the identified nodes in the matrix $\mathbb{M}_{\mathtt{P0}}$ in Figure 4.

$$A_{\mathtt{S2}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \mathcal{F}_X & 0 & 0 & 0 & 0 & \mathcal{F}_H \\ 0 & \mathcal{F}_{P_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{F}_H & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{F}_{P_0} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{F}_{A_+} & 0 \end{pmatrix}$$

A matrix in a QMC is defined with the purpose of evolution of a system, i.e. applying $\mathbb{M}_{\mathtt{P0}}$ on a state vector will give the next state vector in the evolution. However, in our case we are interested in an iterative backward analysis computing in the initial step those states that can reach the goal states in one step and in the $i+1$'th step calculating those states that in one step can reach states that can reach the goal states in i steps.[2] For that kind of calculations the transpose $A^T_{\mathtt{S2}}$ of $A_{\mathtt{S2}}$ is suitable.

$$A^T_{\mathtt{S2}} = \begin{pmatrix} 0 & \mathcal{F}_X & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{F}_{P_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{F}_H & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{F}_{P_0} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{F}_{A_+} \\ 0 & \mathcal{F}_H & 0 & 0 & 0 & 0 \end{pmatrix}$$

We hence aim to solve the equation system

$$A^T_{\mathtt{S2}} |x\rangle + |\psi_{\mathtt{S2}}\rangle = |x\rangle \tag{4}$$

with $|x\rangle = (x_0, x_1, x_3, x_4, x_5, x_7)$ and $|\psi_{\mathtt{S2}}\rangle = (0, \mathcal{F}_{P_{001}}, 0, 0, 0, 0)$. Intuitively the solution to the variable x_i is an operator such that (i, ρ) satisfies $T_{\mathtt{S2}}\, \mathbf{U}\, S_{2,1}$ with probability $Tr(x_i(\rho))$. $P_{001} = |001\rangle \langle 001|$ is the projection leading from $\mathtt{S}_1$ to $S_{2,1}$. The equation system (4) can be written as

$$|x\rangle = (\mathcal{F}_X \circ x_1, \mathcal{F}_{P_1} \circ x_3 + \mathcal{F}_{P_{001}}, \mathcal{F}_H \circ x_4, \mathcal{F}_{P_0} \circ x_5, \mathcal{F}_{A_+} \circ x_7, \mathcal{F}_H \circ x_1)$$

and reduced to

$$\begin{aligned} x_0 &= \mathcal{F}_X \circ x_1 \\ x_1 &= \mathcal{F}_{P_1} \circ \mathcal{F}_H \circ \mathcal{F}_{P_0} \circ \mathcal{F}_{A_+} \circ \mathcal{F}_H \circ x_1 + \mathcal{F}_{P_{001}} \end{aligned}$$

If we did not use the transpose $A^T_{\mathtt{S2}}$ but $A_{\mathtt{S2}}$ instead the equation for x_0 would be $x_0 = 0$ which is not what we are looking for. Recall we did a backward analysis solving the equation system and therefore in the solution to x_1 the operators

[2] In this example we need not consider when states are already a goal state since no initial state is a goal state.

appear in inverse order. Hence letting $\mathcal{E}$ be the operator $\mathcal{E} = \mathcal{F}_H \circ \mathcal{F}_{A_+} \circ \mathcal{F}_{P_0} \circ \mathcal{F}_H \circ \mathcal{F}_{P_1}$ we infer the corresponding recursive continuous endo-function as a solution to x_1

$$\mathcal{F}_{\mathtt{S2}}(\mathcal{F}) = \mathcal{F}_{P_{001}} + \mathcal{F}(\mathcal{E})$$

with least fixed-point

$$\mu\mathcal{F}_{\mathtt{S2}} = \sum_{i=0}^{\infty} \mathcal{F}_{P_{001}}(\mathcal{E}^i)$$

A solution for x_0 and hence the initial state $\mathtt{S0}$ is then $\mathcal{F}_{\mathtt{S0}} = \mu\mathcal{F}_{\mathtt{S2}} \circ \mathcal{F}_X$.

Consider the example $\mathcal{F}_{\mathtt{S0}}(\rho_{000})$ where $\rho_{ijk} = |ijk\rangle \langle ijk|$ then

$$\begin{array}{rcl} \mathcal{F}_{\mathtt{S0}}(\rho_{000}) & = & \mu\mathcal{F}_{\mathtt{S2}}(\rho_{100}) \\ & = & \sum_{i=0}^{\infty} \mathcal{F}_{P_{001}}(\mathcal{E}^i)(\rho_{100}) \\ & = & \sum_{i=0}^{\infty} (\frac{1}{2})^{2(1+4i)} \rho_{001} \\ & = & \frac{64}{255}\rho_{001} \end{array}$$

Hence the probability of $T_{\mathtt{S2}} \,\mathbf{U}\, S_{2,1}$, i.e. the trace of $\mathcal{F}_{\mathtt{S0}}(\rho_{000})$, is $\frac{64}{255} \approx 0.251$.

6.2.3 Computing Constraint Reachability Probabilities

As we saw above, probabilities of constraint reachability events can be analyzed via the least fixed-point of mutual recursive functions over partial density matrices. In this section we prove the correctness of such an approach where the probability of a constrained reachability event is to be within a non-trivial interval determined by a fixed-point.

Given $\mathcal{M} = (\mathbb{M}, \Delta)$ and $T \,\mathbf{U}\, S$ where $S = \cup_b S_{a,b}$ and $T = \cup T_c$. Let $\mathbb{N}$ be $\mathbb{M}$ but where rows and columns with indices not satisfying $T \cup S$ have entries 0 and also $\mathbb{N}_{i,a} = 0$ for all i. Then $\mathbb{N}$ generates simultaneously recursive functions $\mathcal{F}_i : (\mathcal{D} \to \mathcal{D})^n \to (\mathcal{D} \to \mathcal{D})$ defined by

$$\mathcal{F}_i(X_0, \ldots, X_{n-1}) = \begin{cases} I \text{ if } i = a \\ \sum_j X_j \circ \mathbb{N}_{i,j}^T \text{ otherwise} \end{cases}$$

Each $\mathcal{F}_i$ is well-defined since $\sum_j \mathbb{N}_{i,j}^T \leq I$. Furthermore define the endo-function $\mathcal{F}_{S,T}$ on $(\mathcal{D} \to \mathcal{D})^n$ by

$$\mathcal{F}_{S,T}(X_0, \ldots, X_{n-1}) = (\mathcal{F}_0(X_0, \ldots, X_{n-1}), \ldots, \mathcal{F}_{n-1}(X_0, \ldots, X_{n-1}))$$

The lifted partial order $(\mathcal{D} \to \mathcal{D}, \sqsubseteq)$ is a CPO. A Cartesian product of n CPOs is a CPO under componentwise ordering $\sqsubseteq_n$, so $((\mathcal{D} \to \mathcal{D})^n, \sqsubseteq_n)$ is a CPO. All $\mathcal{F}_i$ are continuous if they are continuous in each argument, which they are, and hence $\mathcal{F}_{S,T}$ is continuous and its least fixed-point $\mu F_{S,T}$ is, letting $\bot = (0, \ldots, 0)$, the least upper bound of the increasing sequence

$$\bot \sqsubseteq \mathcal{F}_{S,T}(\bot) \sqsubseteq \ldots \sqsubseteq \mathcal{F}_{S,T}^i(\bot) \sqsubseteq \ldots$$

A solution to X_i is an operator $\mathcal{G}_i$ on $\mathcal{D}$. Intuitively $\mathcal{G}_i(\Delta[i])$ is the sum of those ρ where there exists a path $\pi = (i, \Delta[i])\ldots$ with a prefix $\hat{\pi}$ such that $last(\hat{\pi}) = (a, \rho)$ and for all $\hat{\pi}'$ where $\epsilon < \hat{\pi}' < \hat{\pi}$ then $\hat{\pi}' \not\vdash S$ and $\hat{\pi}' \vdash T$. Hence $Tr(\mathcal{G}_i(\Delta[i]))$ is the probability of reaching a from i through indices satisfying T.

Given the existence of $\mu F_{S,T} = (\mathcal{G}_0, \ldots, \mathcal{G}_{n-1})$ the probability $Pr_{\mathcal{M}}(T \,\mathbf{U}\, S)$ can be bound by the sum of the traces of $(\mathcal{F}_P \circ \mathcal{G}_i)(\Delta[i])$ and $\mathcal{G}_i(\Delta[i])$ respectively where $\mathcal{F}_P = \sum |b\rangle \langle b|$.

Theorem 1 *Let $\mathcal{M} = (\mathbb{M}, \Delta)$, $S = \cup_b S_{a,b}$, and $T = \cup T_c$, and let $\mu F_{S,T} = (\mathcal{G}_0, \ldots, \mathcal{G}_{n-1})$ and $\mathcal{F}_P = \sum |b\rangle \langle b|$ then*

$$p_0 \leq Pr_{\mathcal{M}}(T \,\mathbf{U}\, S) \leq p_1$$

where $p_0 = \sum_i Tr((\mathcal{F}_P \circ \mathcal{G}_i)(\Delta[i]))$ and $p_1 = \sum_i Tr(\mathcal{G}_i(\Delta[i]))$.

Proof Let $\mathcal{M} = (\mathbb{M}, \Delta)$, $S = \cup_b S_{a,b}$, $T = \cup T_c$, and let $\mu F_{S,T} = (\mathcal{G}_0, \ldots, \mathcal{G}_{n-1})$ and $\mathcal{F}_P = \sum |b\rangle \langle b|$. Due to Equation (1) and (3) it is sufficient to prove

$$\sum_i Tr((\mathcal{F}_P \circ \mathcal{G}_i)(\Delta[i])) \leq \sum_{\hat{\pi} \in Path_{\mathcal{M}}(T \,\mathbf{U}\, S)} Tr(last(\hat{\pi})) \leq \sum_i Tr(\mathcal{G}_i(\Delta[i]))$$

where $Tr(last(\hat{\pi})) = Tr(\rho)$ when $last(\hat{\pi}) = (j, \rho)$ for some j. Letting

$$\begin{aligned} \mathcal{F}_i^0(X_0, \ldots, X_{n-1}) &= X_i \\ \mathcal{F}_i^{j+1}(X_0, \ldots, X_{n-1}) &= \mathcal{F}_i(\mathcal{F}_0^j(X_0, \ldots, X_{n-1}), \ldots, \mathcal{F}_{n-1}^j(X_0, \ldots, X_{n-1})) \end{aligned}$$

then since for $j > 0$

$$\mathcal{F}_{S,T}^j(\bot) = (\mathcal{F}_0(\mathcal{F}_0^{j-1}(\bot), \ldots, \mathcal{F}_{n-1}^{j-1}(\bot)), \ldots, \mathcal{F}_{n-1}(\mathcal{F}_0^{j-1}(\bot), \ldots, \mathcal{F}_{n-1}^{j-1}(\bot)))$$

$\mathcal{G}_i$ is the least upper bounds of the increasing sequence

$$\{\mathcal{G}_i^j = \mathcal{F}_i(\mathcal{F}_0^{j-1}(\bot), \ldots, \mathcal{F}_{n-1}^{j-1}(\bot))\}$$

so it is sufficient to prove for all j that

$$\sum_i Tr((\mathcal{F}_P \circ \mathcal{G}_i^j)(\Delta[i])) \leq \sum_{\hat{\pi} \in Path_{\mathcal{M}}(T \,\mathbf{U}^{\leq j}\, S)} Tr(last(\hat{\pi})) \tag{5}$$

and

$$\sum_{\hat{\pi} \in Path_{\mathcal{M}}(T \,\mathbf{U}^{\leq j}\, S)} Tr(last(\hat{\pi})) \leq \sum_i Tr(\mathcal{G}_i^j(\Delta[i])) \tag{6}$$

Notice that $\mathcal{G}_i^j$ is inductively defined by

$$\mathcal{G}_i^1 = \begin{cases} I \text{ if } i = a \\ 0 \text{ otherwise} \end{cases}$$

$$\mathcal{G}_i^{j+1} = \begin{cases} I \text{ if } i = a \\ \sum_k \mathcal{G}_k^j \circ \mathbb{N}_{i,k}^T \text{ otherwise} \end{cases}$$

We first prove (6). Assume $\hat{\pi} \in Path_{\mathcal{M}}(T\,\mathbf{U}^{\leq j}\,S)$. Then for some $i_1, \ldots, i_{|\hat{\pi}|}$ with $|\hat{\pi}| \leq j$

$$\hat{\pi} = (i_1, \rho_1) \ldots (i_{|\hat{\pi}|}, \rho_{|\hat{\pi}|})$$

where $\rho_1 = \Delta[i_1]$, and $\rho_{k+1} = \mathbb{M}_{i_{k+1},i_k}(\rho_k)$. Also $(i_{|\hat{\pi}|}, \rho_{|\hat{\pi}|}) \vdash S$ so $i_{|\hat{\pi}|} = a$ and when $k < |\hat{\pi}|$ then $(i_k, \rho_k) \vdash T$ and $(i_k, \rho_k) \nvdash S$. Hence $\mathbb{N}_{i_{k+1},i_k} = \mathbb{M}_{i_{k+1},i_k}$ so there exists

$$\mathcal{F}^{i_1}_{\hat{\pi}} = I \circ \mathbb{N}^T_{i_{|\hat{\pi}|-2},i_{|\hat{\pi}|-1}} \circ \cdots \circ \mathbb{N}^T_{i_1,i_2}$$

such that $Tr(\mathcal{F}^{i_1}_{\hat{\pi}}(\rho_1)) > 0$. Therefore, for any $\hat{\pi} = (i, \Delta[i]) \ldots \in Path_{\mathcal{M}}(T\,\mathbf{U}^{\leq j}\,S)$ there exists an operator $\mathcal{F}^i_{\hat{\pi}}$ with $\mathcal{F}^i_{\hat{\pi}}(\Delta[i]) = \rho$, $last(\hat{\pi}) = (a, \rho)$. Then letting

$$Path^i_{\mathcal{M}}(T\,\mathbf{U}^{\leq j}\,S) = \{\hat{\pi} = (i, \Delta[i]) \ldots \mid \hat{\pi} \in Path_{\mathcal{M}}(T\,\mathbf{U}^{\leq j}\,S)\}$$

by definition of $\mathcal{G}^j_i$ we have for all i

$$\sum_{\hat{\pi} \in Path^i_{\mathcal{M}}(T\,\mathbf{U}^{\leq j}\,S)} Tr(\mathcal{F}^i_{\hat{\pi}}(\Delta[i])) \leq \sum_i Tr(\mathcal{G}^j_i(\Delta[i]))$$

from which we infer (6). Next we prove (5). If $\sum_i Tr((\mathcal{F}_P \circ \mathcal{G}^j_i)(\Delta[i])) = 0$ we are done. Suppose instead $Tr((\mathcal{F}_P \circ \mathcal{G}^j_i)(\Delta[i])) > 0$. By definition $\mathcal{F}_P \circ \mathcal{G}^j_i = \sum_k \mathcal{E}_k$ where each $\mathcal{E}_k$ is on the form

$$\mathcal{E}_k = \mathcal{F}_P \circ I \circ \mathbb{N}^T_{i_{l-1},i_l} \circ \cdots \circ \mathbb{N}^T_{i_1,i_2}$$

where $l < j$. Also, for each $\mathcal{E}_k$ where $Tr(\mathcal{E}_k(\Delta[i])) > 0$ there exists

$$\hat{\pi}_k = (i_1, \rho_1) \ldots (i_l, \rho_l)$$

where $i_1 = i$, $\rho_1 = \Delta[i]$, $\rho_{m+1} = \mathbb{M}_{i_{m+1},i_m}(\rho_m)$, $Tr(\mathcal{F}_P(\rho_l)) > 0$. Also $i_l = a$ and $i_m \vdash T$, $i_m \nvdash S$ for all $m < l$. Therefore $\hat{\pi}_k \vdash S$ and $\forall \epsilon < \hat{\pi} < \hat{\pi}_k . \hat{\pi} \vdash T, \hat{\pi} \nvdash S$ and hence $\hat{\pi}_k \in Path^i_{\mathcal{M}}(T\,\mathbf{U}^{\leq j}\,S)$. It then follows that for all i

$$Tr((\mathcal{F}_P \circ \mathcal{G}^j_i)(\Delta[i])) \leq \sum_{\pi \in Path^i_{\mathcal{M}}(T\,\mathbf{U}^{\leq j}\,S)} Tr(last(\hat{\pi}))$$

from which we infer (5). □

Needless so say, the probability intervals of bounded constrained rechability events $T\,\mathbf{U}^{\leq i}\,S$ may be computed by fixed-point approximations $\mathcal{F}^i_{S,T}(\bot)$.

On a final note, why an interval and not an exact probability in Theorem 1? Reconsider the example from Section 6.2.2 where we calculated the probability of reaching a certain state where the register value is 1. In the example the register always possesses just a single value, it is never in a superposition having several values simultaneously with some probability. Computing the probability is a matter of reachability of path states $(2, \rho)$ where $Tr(\mathcal{F}_{P_{001}}(\rho)) > 0$. For the successful state we have $Tr(\mathcal{F}_{P_{001}}(\rho)) = Tr(\rho)$. However, commonly for $S = \cup_b S_{a,b}$ and $P = \sum_b |b\rangle \langle b|$ we only know $Tr(\mathcal{F}_P(\rho)) \leq Tr(\rho)$. Therefore we cannot in general compute the precise value of $Pr_{\mathcal{M}}(T\,\mathbf{U}\,S)$, we can only constrain it by an interval.

7 Conclusion

We have presented a simple imperative quantum pseudo programming language with a denotational semantics similar to [10] that leads naturally to our definition of a QMC. The probability measure of a QMC is defined similar to the probability measure for a DTMC in contrast to a super-operator valued measure as in [2]. Our main novel contribution is the demonstration of how intervals for probabilities for bounded reachability events may be computed as transient state probabilities and how probability intervals for unbounded reachability events may be computed as the least fixed-point of a set of linear equations over operators on partial density matrices.

Our notation for constraint reachability events allows only formulas on the form $\cup T_i \, \mathbf{U} \cup_j S_{k,j}$ where k is fixed, so as future work it would be interesting to extend our framework to model checking properties in quantum extensions of e.g. CTL and LTL. How would e.g. model checking in our framework compare with the algorithms presented in [11]? Also, it would be interesting to pursue a refinement of the QMC probability space to allow for exact calculation of probabilities instead of just intervals. Finally, we plan to investigate under which conditions a QMC has a finite number of path states and investigate how such finiteness may influence the algorithmic complexity of model checking.

References

[1] C. Baier and J.P. Katoen. *Principles of Model Checking*. MIT Press, 2008. ISBN: 9780262026499.

[2] Y. Feng, N. Yu, and M. Ying. "Model checking quantum Markov chains". In: *Journal of Computer and System Sciences* 79.7 (2013), pp. 1181–1198. ISSN: 0022-0000. DOI: https://doi.org/10.1016/j.jcss.2013.04.002.

[3] R. P. Feynman. "Simulating Physics with Computers". In: *International Journal of Theoretical Physics* 21 (1982), pp. 467–488. DOI: 10.1007/BF02650179.

[4] S. Gay, R. Nagarajan, and N. Papanikolaou. *Probabilistic Model–Checking of Quantum Protocols*. 2005. arXiv: quant-ph/0504007 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0504007.

[5] S. Gudder. "Quantum Markov chains". In: *Journal of Mathematical Physics* 49.7 (July 2008), p. 072105. ISSN: 0022-2488. DOI: 10.1063/1.2953952.

[6] M. Kwiatkowska, G. Norman, and D. Parker. "PRISM 4.0: Verification of Probabilistic Real-Time Systems". In: *Computer Aided Verification*. Springer Berlin Heidelberg, 2011, pp. 585–591. ISBN: 978-3-642-22110-1.

[7] B. Ömer. "Structured Quantum Programming". PhD thesis. Institute for Theoretical Physics, Vienna University of Technology, 2003. URL: http://tph.tuwien.ac.at/~oemer/doc/structquprog.pdf.

[8] P. Selinger. "Towards a Quantum Programming Language". In: *Mathematical Structures in Computer Science* 14.4 (2004), pp. 527–586. DOI: 10.1017/S0960129504004256.

[9] P. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[10] M. Ying. "Floyd-hoare logic for quantum programs". In: *ACM Trans. Program. Lang. Syst.* 33.6 (2011), pp. 1–49. DOI: 10.1145/2049706.2049708.

[11] M. Ying and Y. Feng. *Model Checking Quantum Systems: Principles and Algorithms*. Cambridge University Press, 2021. ISBN: 9781108484305.