QUANTUM CIRCUITS FOR tnreason

RESEARCH NOTES IN THE ENEXA AND QROM PROJECTS

October 30, 2025

By its central axioms, quantum mechanics of multiple qubits is formulated by tensors capturing states and discrete time evolutions. Motivated by the structural similarity, we investigate how quantum circuits can be utilized for the tensor-network based approach towards efficient and explainable AI in the `tnreason` framework.

We follow two main ideas:

- **Sampling of Computation-Activation Networks:** Prepare quantum states, which measurement statistics can be utilized to prepare samples from Computation-Activation Networks.
- **Quantum Circuits as Contraction providers:** Quantum circuits are contractions of multiple tensors and therefore tensor networks, and measurement probabilities are given by contractions. Here we investigate how we can exploit these as contraction provider for `tnreason`.

Contents

1 Quantum Computation Basics

1.1 State Encoding Schemes

Basis Encoding in Quantum Computation refers to the representation of classical n bit strings by n qubit basis states, and is called one-hot encoding in $\mathrm{tnreason}$. The Basis Encoding scheme in $\mathrm{tnreason}$ goes beyond this scheme and also encodes subsets by sums of one-hot encodings to the members of the set. In this way, relations and functions are represented by boolean tensors and contraction of them is refered as Basis calculus.

Amplitude Encoding in Quantum Computation refers to the storage of complex numbers in the amplitudes of quantum states. The pendant in $\mathrm{tnreason}$ is the Coordinate Encoding scheme, where real numbers are stored in the coordinates of real-valued tensors. Compared to Amplitude Encoding, Coordinate Encoding does not have the normalization constraint of quantum states. The Amplitude Encoding of the square root of a probability distribution is sometimes called q-sample.

1.2 Controlled Single Qubit Gates

We define the rotation gate around the Y-axis by an angle α as

$$R_Y(\alpha)\left[A_{\mathrm{in}}, A_{\mathrm{out}}\right] := \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) & -\sin\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) & \cos\left(\frac{\alpha}{2}\right) \end{bmatrix}$$

Further we define the Pauli-X:

$$\sigma_1\left[A_{\mathrm{in}}, A_{\mathrm{out}}\right] := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Controlled single qubit gates are defined using control qubits, where the gate is applied to the target qubit if the control qubits are in a specific state and the identity is applied otherwise. In the tensor network diagrams, we do not distinguish between incoming and outgoing control qubit variables, since the control acts as a Dirac tensor. Thus, controlled unitary with target qubit X_t and control qubits X_c are represented by tensors

$$\mathcal{U}\left[X_{t,\mathrm{in}}, X_{t,\mathrm{out}}, X_c\right]$$

where for each state x_c to the control variables we have that

$$\mathcal{U}\left[X_{t,\mathrm{in}}, X_{t,\mathrm{out}}, X_c = x_c\right]$$

is a unitary matrix acting on the leg space of the target variable.

1.3 Measurement and Phases

The computational basis measurement of the qubits X_A of a Quantum State $\psi\left[X_{[d]}\right]$ is equal to drawing samples from a distribution

$$\mathbb{P}\left[X_A\right] = \left\langle \psi\left[X_{[d]}\right], \psi^*\left[X_{[d]}\right] \right\rangle \left[X_A\right] .$$

Here $\psi^*\left[X_{[d]}\right]$ is the complex conjugate of $\psi\left[X_{[d]}\right]$. When ψ is prepared by a quantum circuit acting on a initial state, the complex conjugate is the hermitean conjugate of the circuit acting on the complex conjgate of the initial state.

We abbreviate these contractions by extending the contraction diagrams with measurement symbols (see Figure 1).

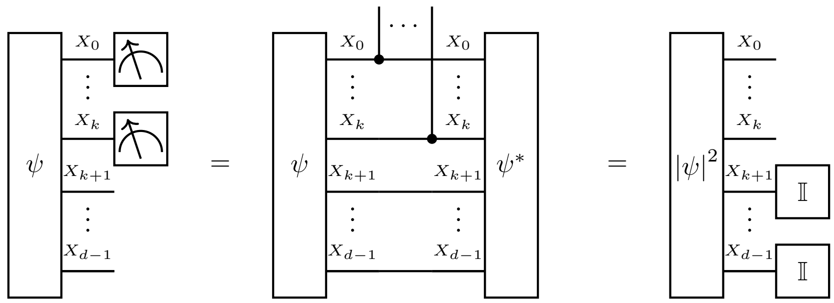


Figure 1: Computational Basis Measurement of a quantum state ψ. The measurement symbols on the left side indicate the measured qubits and the first equation is understood as a definition. In the second equation we sketch, that the measurement distribution is equal to the contraction of the square absolute transform of ψ to the measured variables.

Each complex-valued tensor $\psi\left[X_{[d]}\right]$ has a decomposition into a phase tensor $\phi\left[X_{[d]}\right]$ and an absolute tensor $|\psi|\left[X_{[d]}\right]$ defined by

$$\psi\left[X_{[d]}\right] = \left\langle \exp\left[i \cdot \phi\left[X_{[d]}\right]\right], |\psi|\left[X_{[d]}\right]\right\rangle \left[X_{[d]}\right] .$$

The measurement distribution is depends only on $|\phi|$, that is

$$\mathbb{P}\left[X_{[d]}\right] = |\psi|^2\left[X_{[d]}\right].$$

Note, that when only a subset of variables is measured, the distribution is the contraction of the absolute square transform (these operations do not commute)

$$\mathbb{P}\left[X_A\right] = \left\langle |\psi|^2\left[X_{[d]}\right]\right\rangle\left[X_A\right] .$$

When we are interested in the preparation of quantum states with a specific computational basis measurement distribution, we can restrict to states with vanishing phase cores, that is

$$\psi\left[X_{[d]}\right] = \left\langle \exp\left[i \cdot 0\left[X_{[d]}\right]\right], |\psi|\left[X_{[d]}\right]\right\rangle\left[X_{[d]}\right] = |\psi|\left[X_{[d]}\right].$$

1.4 Graph-Controlled circuits

Definition 1 (Graph-Controlled Circuit)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic hypergraph, where each hyperedge has exactly one outgoing node and all nodes appear exactly once as outgoing nodes of an hyperedge. Then a by $\mathcal{G}$ controlled circuit is a decoration of the edges $e = (\mathcal{V}^{\mathrm{in}}, \{v\}) \in \mathcal{E}$ by controlled unitaries*

$$\mathcal{U}^e\left[X_{v,\mathrm{in}}, X_{v,\mathrm{out}}, X_{\mathcal{V}^{\mathrm{in}},\mathrm{out}}\right] .$$

Theorem 1. *Let $\mathcal{G}$ be a directed acyclic graph. The measurement distributions of the by $\mathcal{G}$ controlled circuits acting on disentangled initial states are equal to the Bayesian Networks on $\mathcal{G}$.*

Lemma 1. *Any Bayesian network on a directed acyclic graph $\mathcal{G}$ can be prepared by a $\mathcal{G}$-controlled circuit with activation circuits of the conditional probability tensors.*

Proof. Let $\mathbb{P}\left[X_{[d]}\right]$ be a Bayesian network on the graph $\mathcal{G}$. Enumerate the nodes $\mathcal{V}$ of the $\mathcal{G}$ by $[d]$, such that for each $k \in [d]$ we have $\mathrm{Pa}(k) \subset [k]$. Then define a $\mathcal{G}$-controlled circuit, by choosing for each $k \in [d]$ controlled unitaries which satisfy

$$\mathcal{U}^k\left[X_{k,\mathrm{in}} = 0, X_{v,\mathrm{out}}, X_{\mathrm{Pa}(k),\mathrm{out}}\right] = \sqrt{\mathbb{P}\left[X_k | X_{\mathrm{Pa}(k)}\right]} .$$

Here we specified only the action of the controlled unitary on the basis vector $\epsilon_0\left[X_k\right]$, the action on $\epsilon_1\left[X_k\right]$ can be chosen by an arbitrary orthogonal unit vector. Any such defined $\mathcal{G}$-controlled circuit acting on the initial state $\bigotimes_{k\in[d]} \epsilon_0\left[X_k\right]$ prepares a quantum state $\psi\left[X_{[d]}\right]$ with measurement distribution

$$|\psi|^2\left[X_{[d]}\right] .$$

Given arbitrary $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$ we have

$$|\psi|^2 \left[X_{[d]} = x_{[d]}\right] = \prod_{k\in[d]} \mathbb{P}\left[X_k = x_k \middle| X_{\mathrm{Pa}(k)} = x_{\mathrm{Pa}(k)}\right] = \mathbb{P}\left[X_{[d]} = x_{[d]}\right] .$$

Here we used in the last equation, that $\mathbb{P}\left[X_{[d]}\right]$ is a Bayesian network. Since the equivalence holds for any coordinate, this establishes the equivalence of the measurement distribution of $\psi\left[X_{[d]}\right]$ and $\mathbb{P}\left[X_{[d]}\right]$. ☐

Lemma 2. *Let $(\mathcal{G},\mathcal{U})$ be a $\mathcal{G}$-controlled circuit acting on a disentangled initial state and $\mathbb{P}\left[X_{\mathcal{V}}\right]$ the corresponding measurement distribution. Then we have for each $v \in \mathcal{V}$ the conditional independence*

$$\left(X_v \perp X_{\mathrm{NonDes}(v)}\right) \big| \, X_{\mathrm{Pa}(v)} .$$

Proof. We choose to a given $v \in \mathcal{V}$ an enumeration $[d]$ of the nodes, such that for each $k \in [d]$ we have $\mathrm{Pa}(k) \subset [k]$ and for the enumerator $\tilde{k}$ of v we further have $\mathrm{NonDes}(\tilde{k}) \subset [\tilde{k}]$. Let $\mathbb{P}\left[X_{[d]}\right]$ be the measurement distribution of the $\mathcal{G}$-controlled circuit acting on a disentangled initial state $\bigotimes_{k\in[d]} \psi^k\left[X_k\right]$ and choose arbitrary $x_{[k]}$. We then have

$$\begin{aligned}
\mathbb{P}\left[X_{\tilde{k}}, X_{[k]} = x_{[k]}\right] &= \left\langle \left(\bigcup_{k\in[d]} \{\mathcal{U}^k,\mathcal{U}^{k,\dagger},\psi^k,\psi^{k,*}\}\right) \cup \left(\bigcup_{k\in[\tilde{k}]} \epsilon_{x_k}\left[X_{k,\mathrm{out}}\right]\right) \right\rangle \left[X_{\tilde{k}}\right] \\
&= \left\langle \bigcup_{k\in[\tilde{k}]} \{\mathcal{U}^k,\mathcal{U}^{k,\dagger},\psi^k,\psi^{k,*},\epsilon_{x_k}\left[X_{k,\mathrm{out}}\right]\} \right\rangle \left[X_{\tilde{k}}\right] \\
&= \left| \left\langle \mathcal{U}^{\tilde{k}}\left[X_{\tilde{k},\mathrm{in}}, X_{\tilde{k},\mathrm{out}}\right], X_{\mathrm{Pa}(\tilde{k}),\mathrm{out}} = x_{\mathrm{Pa}(\tilde{k}),\mathrm{out}} \right\rangle \left[X_{\tilde{k},\mathrm{out}}\right] \right|^2 \\
&\quad \cdot \prod_{k\in[\tilde{k}]} \left(\left\langle \psi^k\left[X_{k,\mathrm{in}}\right], \mathcal{U}^k\left[X_{k,\mathrm{in}}, X_{k,\mathrm{out}}, X_{\mathrm{Pa}(k),\mathrm{out}} = x_{\mathrm{Pa}(k),\mathrm{out}}\right] \right\rangle \left[\varnothing\right]\right)^2
\end{aligned}$$

Here we used in the second equation the unitarity of the controlled unitaries to $k \notin [\tilde{k}]$. Since the indices $x_{[\tilde{k}]/\mathrm{Pa}(\tilde{k})} = x_{\mathrm{NonDes}(\tilde{k})}$ appear only in the constant term, we conclude

$$\mathbb{P}\left[X_{\tilde{k}} \middle| X_{[k]}\right] = \mathbb{P}\left[X_{\tilde{k}} \middle| X_{\mathrm{Pa}(\tilde{k})}\right] \otimes \mathbb{I}\left[X_{\mathrm{NonDes}(\tilde{k})}\right] ,$$

which establishes the conditional independence $\left(X_v \perp X_{\mathrm{NonDes}(v)}\right) \big| \, X_{\mathrm{Pa}(v)}$. ☐

Proof of Thm. 1. The theorem follows directly from the two lemmas, using that Bayesian Networks are characterized by the conditional independence of each variable to its non-descendants given its parents. ☐

Another question is, whether each quantum state, which measurement distribution is a Bayesian Network can be prepared by a $\mathcal{G}$-controlled circuit. This is not the case, since the phase tensor of a by $\mathcal{G}$-controlled circuit has a decomposition

$$\phi\left[X_{[d]}\right] = \sum_{k\in[d]} \phi^k\left[X_k, X_{\mathrm{Pa}(k)}\right] \otimes \mathbb{I}\left[X_{[d]/\{\{k\}\cup\mathrm{Pa}(k)\}}\right] ,$$

where the phase cores ϕ^k can be read of the controlled unitaries.

2 Circuit Encoding Schemes

We investigate here quantum pendants to the function encoding schemes used in $\mathrm{tnreason}$.

- Pendant for Coordinate Encoding in $\mathrm{tnreason}$: Amplitude Encoding, storing the function value in the amplitude of an ancilla qubit. This is realized by an **Activation circuit**.
- Pendant for Basis Encoding in $\mathrm{tnreason}$: **Computation circuit**, with composition by contraction property.

Both are defined using controlled single qubit gates (see Sections 4.2-3 in [Nielsen, Chuang]) with ancilla qubits being the target qubits.

2.1 Activation circuit

We define the angle preparing function on $p \in [0,1]$ by

$$h(p) = 2 \cdot \cos^{-1}\left(\sqrt{1-p}\right) .$$

For any $p \in [0,1]$ we then have

$$\langle \epsilon_0 \left[A_{\mathrm{in}}\right], R_Y\left(h(p)\right)\left[A_{\mathrm{in}}, A_{\mathrm{out}}\right] \rangle \left[A_{\mathrm{out}}\right] = \begin{bmatrix} \sqrt{1-p} \\ \sqrt{p} \end{bmatrix} .$$

Definition 2 (Activation circuit). *Given a function*

$$q : \bigtimes_{l \in [p]} [2] \to [0,1]$$

its activation circuit is the controlled unitary $\mathcal{V}^{\tau}\left[A_{\mathrm{in}}, A_{\mathrm{out}}, Y_{[p]}\right]$ *defined as*

$$\mathcal{V}^{\tau}\left[A_{\mathrm{in}}, A_{\mathrm{out}}, Y_{[p]}\right] = \sum_{y_{[p]}} \epsilon_{y_{[p]}}\left[Y_{[p]}\right] \otimes R_Y\left(h(q(y_{[p]}))\right)\left[A_{\mathrm{in}}, A_{\mathrm{out}}\right] .$$

We will ease our notation by dropping the in and out labels to the control variables. This amounts to understanding the Dirac delta tensors in activation circuits as hyperedges. Along that picture the quantum circuit is a tensor network on hyperedges instead of edges.

When we have a probability tensor, its activation circuit be prepared, since all values are in $[0,1]$. Note that for rejection sampling, only the quotients of the values are important, we can therefore scale the value by a scalar such that the mode is 1.

Tensors $\tau\left[Y_{[p]}\right]$ with non-negative coordinates can be encoded after dividing them by their maximum, that is the activation circuit of the function

$$q(y_{[p]}) = \frac{\tau\left[Y_{[p]} = y_{[p]}\right]}{\max_{\tilde{y}_{[p]}} \tau\left[Y_{[p]} = \tilde{y}_{[p]}\right]}$$

When the maximum of the tensor is not known, it can be replaced by an upper bound (reducing the acceptance rate of the rejection sampling).

2.1.1 Encoding of directed tensors

Following the schemes in **?**, we can prepare the acyclic networks of directed and non-negative tensors by a sequence of controlled rotations. Directed and non-negative tensors correspond with conditional probability distributions and acyclic networks are Bayesian Networks. We prepare them by activation circuits of functions (see Figure 2)

$$x_{\mathrm{Pa}(k)} \to \mathbb{P}\left[X_k = 1 \middle| X_{\mathrm{Pa}(k)} = x_{\mathrm{Pa}(k)}\right] .$$

In this way, Bayesian Networks can be prepared as quantum circuits, where each conditional probability distribution is prepared by an activation circuit.

Theorem 2 (Low et al.). *Any Bayesian Network of variables $X_{[d]}$, where the enumeration by $[d]$ respects the partial order by child-parent relations, can be prepared as a quantum circuit by concatenating the activation circuits*

$$\mathcal{V}^{\mathbb{P}\left[X_k=1 \middle| X_{\mathrm{Pa}(k)}\right]}\left[X_{k,\mathrm{in}}, X_k, X_{\mathrm{Pa}(k)}, X_{\mathrm{Pa}(k)}\right]$$

for $k \in [d]$ and acting on the initial state $\bigotimes_{k \in [d]} \epsilon_0\left[X_{k,\mathrm{in}}\right]$.

2.2 Computation circuits

We here suggest a quantum pendant to basis encodings (see Chapter Basis Calculus), which has the decomposition by contraction property.

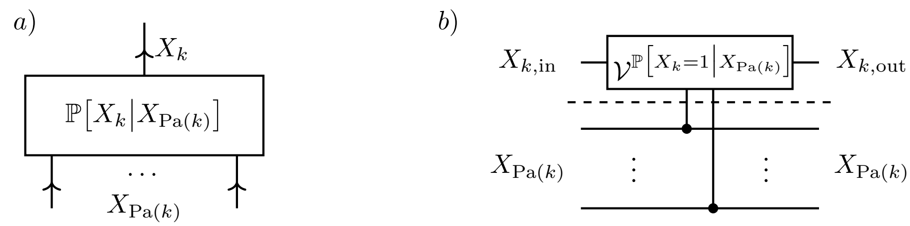


Figure 2: Representation of directed and positive tensor by a controlled rotation. a) Conditional probability tensor $\mathbb{P}\left[X_k \middle| X_{\mathrm{Pa}(k)}\right]$ being a tensor in a Bayesian Network. b) Circuit Encoding as a controlled rotation, which is the Activation circuit of the tensor $\mathbb{P}\left[X_k = 1 \middle| X_{\mathrm{Pa}(k)}\right]$.

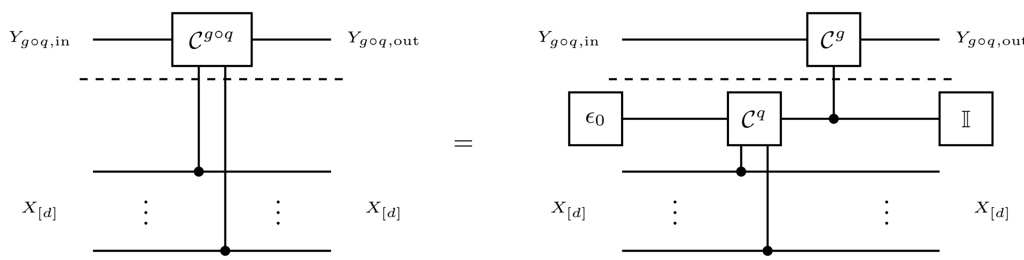


Figure 3: Exploitaition of Decomposition sparsity in computation circuits.

Definition 3 (Computation circuit)**.** *Given a boolean function $q : \times_{k\in[d]}[2] \to [2]$ the computation circuit is the unitary tensor*

$$\begin{aligned}\mathcal{C}^q\left[Y_{\mathrm{in},q}, Y_{\mathrm{out},q}, X_{[d]}\right] \quad = & \sum_{x_{[d]}\in\times_{k\in[d]}[m_k]\,:\,q(x_{[d]})=1} \sigma_1\left[Y_{\mathrm{in},q}, Y_{\mathrm{out},q}\right] \otimes \epsilon_{x_{[d]}}\left[X_{[d]}\right] \\ & + \sum_{x_{[d]}\in\times_{k\in[d]}[m_k]\,:\,q(x_{[d]})=0} \delta\left[Y_{\mathrm{in},q}, Y_{\mathrm{out},q}\right] \otimes \epsilon_{x_{[d]}}\left[X_{[d]}\right] .\end{aligned}$$

Notice, that $U^{\neg} = \mathrm{CNOT}$, which is obvious from $\mathbb{I}\left[Y_{\mathrm{in}}, Y_{\mathrm{out}}\right] - \delta\left[Y_{\mathrm{in}}, Y_{\mathrm{out}}\right]$ being the Pauli-X gate (not to be confused with X denoting distributed variables here). The computation circuit is therefore a generalized controlled NOT gate, where the control is by a boolean function.

Functions with multiple output variables, i.e. $q : \times_{k\in[d]}[2] \to \times_{l\in[p]}[2]$, can be encoded image coordinate wise as a concatenation of the respective circuits.

2.2.1 Composition by Contraction - Exploiting Decomposition sparsity

The decomposition by contraction property of basis encodings is now a composition of circuits property, as stated in the next lemma.

Lemma 3. *We have for functions $q : \times_{k\in[d]}[2] \to \times_{l\in[p]}[2]$, $g : \times_{l\in[p]}[2] \to \times_{s\in[r]}[2]$ (see Figure 3)*

$$\begin{aligned}\mathcal{C}^{g\circ q}\left[Y_{\mathrm{in},g\circ q}, Y_{\mathrm{out},g\circ q}, X_{\mathrm{in},[d]}, X_{\mathrm{out},[d]}\right] = \Big\langle & \epsilon_0\left[Y_{\mathrm{in},q}\right], \\ & \mathcal{C}^q\left[Y_{\mathrm{in},q}, Y_{\mathrm{out},q}, X_{\mathrm{in},[d]}, X_{\mathrm{out},[d]}\right], \\ & \mathcal{C}^g\left[Y_{\mathrm{in},g\circ q}, Y_{\mathrm{out},g\circ q}, Y_{\mathrm{out},q}, Y_{\mathrm{out},q}\right]\Big\rangle\left[Y_{\mathrm{in},g\circ q}, Y_{\mathrm{out},g\circ q}, X_{\mathrm{in},[d]}, X_{\mathrm{out},[d]}\right] .\end{aligned}$$

When having a syntactical decomposition of a propositional formula, we can iteratively apply the computation circuit decomposition theorem and prepare each connective by a circuit. We can decompose any propositional formula into logical connectives and prepare to each a modulus 2 circuit implementation. This works, when the target qubit of one connective is used as a value qubit of another.

Note, that the variables $Y_{\mathrm{out,aux}}$ are auxiliar and not left open in the contraction. This amounts to not measuring them in a computational basis measurement.

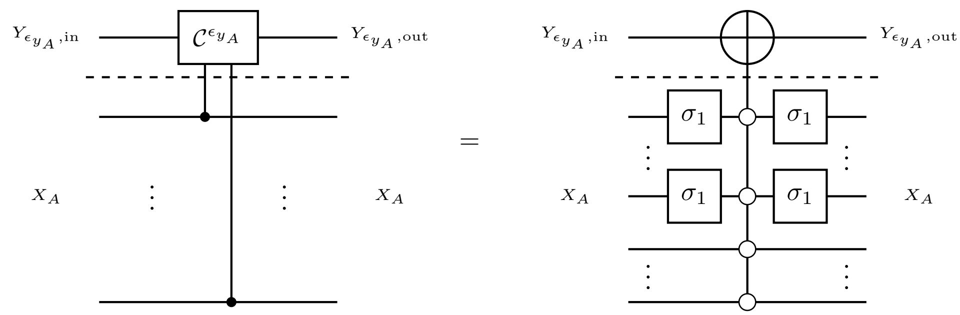


Figure 4: Exploitaition of Polynomial sparsity in computation circuits. Here, we use the typical denotation of a muliple-controlled CNOT gate, which control symbols do not indicate Dirac tensors.

2.2.2 Construction for mod2-basis+ CP decompositions - Exploiting Polynomial sparsity

Concatenating two computation circuits, which have the same head qubit, is the computation circuit of their mod2 sum.

A basis+ elementary function can be encoded by a single controlled NOT operation with auxiliary X qubits.

This motivates the mod2-basis+ CP decomposition of tensors, which is exactly the decomposition of boolean polynomials into monomials. Each monomial is called a terms (products of x or (1-x) factors), and minterms in case that all variables appear.

Definition 4. *Given a boolean tensor τ, a mod2-basis+ CP decomposition is a collection $\mathcal{M}$ of tuples (A, x_A) with such that for any $x_{[d]} \in \bigtimes_{k\in[d]}[m_k]$*

$$\tau\left[X_{[d]} = x_{[d]}\right] = \bigoplus_{(A,x_A)\in\mathcal{M}} \langle \epsilon_{x_A}\left[X_A\right]\rangle \left[X_{[d]} = x_{[d]}\right] .$$

Using that basis CP decompositions are a special case of basis+ CP decompositions, we get the following rank bound.

Lemma 4. *The mod2-basis+ CP rank is bounded by the basis CP rank.*

Proof. Use $A = [d]$, and x_A to each supported state. Then the mod2-sum is a usual sum and the basis CP decomposition is also a mod2-basis+ CP decomposition. □

This shows in particular, that any propositional formula can be represented by a mod2-basis+ CP decomposition.

Lemma 5. *The computation circuit to a boolean tensor τ with a mod2-basis+ CP decomposition $\mathcal{M}$ obeys*

$$\begin{aligned}
&\mathcal{C}^{\tau}\left[Y_{\tau,\mathrm{in}}, Y_{\tau,\mathrm{out}}, X_{[d],\mathrm{in}}, X_{[d],\mathrm{in}}\right] \\
&\quad = \left\langle \left\{\delta\left[Y_{\tau,\mathrm{in}}, Y_0\right], \delta\left[Y_{\tau,\mathrm{out}}, Y_{|\mathcal{M}|-1}\right]\right\} \cup \left\{\mathcal{C}^{\epsilon_{x_A}}\left[Y_i, Y_{i+1}, X_{A,\mathrm{in}}, X_{A,\mathrm{out}}\right] \,:\, (A, x_A)\in\mathcal{M}\right\}\right\rangle \left[Y_{\tau,\mathrm{in}}, Y_{\tau,\mathrm{out}}, X_{[d],\mathrm{in}}, X_{[d],\mathrm{out}}\right]
\end{aligned}$$

where $i \in [|\mathcal{M}|]$ enumerates the tuples in $\mathcal{M}$.

Each computation circuit to each boolean monomial can be prepared by a multiple-controlled σ_1 gate and further pairs of σ_1 gates preparing the control state, see Figure 4. When we sum monomials wrt modulus 2 calculus, then the preparation is a sequence of such circuits. In such way, we can prepare the computation circuit to any propositional formula. This encoding strategy exploits a modified (by mod2 calculus) polynomial sparsity.

2.2.3 Preparation by fine and coarse structure

Having a mod2-basis+ CP decomposition of rank r to a connective, we need r controlled NOT gates to prepare the basis encoding. Given a syntactical decomposition of a boolean statistics, we prepare the basis encoding as a circuit with:

- **Fine Structure:** Represent each logical connective based on its mod2-basis+ CP decomposition, as a concatenation of computation circuits with the same variables.
- **Coarse Structure:** Arrange the logical connective representing circuits according to the syntactical hypergraph, where parent head variables appear as distributed variables at their children.

3 Preparation of Distributions

We investigate, how the above circuit encoding schemes can be applied in the preparation of states, which computational basis measurements are samples from specific distributions.

3.1 Generic Q-samples

In general, we define Q-samples to be quantum states, which measured in the computational basis reproduce a given probability distribution.

Definition 5 (Q-sample)**.** *Given a probability distribution $\mathbb{P} : \bigtimes_{k\in[d]}[2] \to \mathbb{R}$ (i.e. $\langle\mathbb{P}\rangle[\varnothing] = 1$ and $0 \prec \mathbb{P}$) its q-sample is*

$$\psi^{\mathbb{P}}\left[X_{[d]}\right] = \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \sqrt{\mathbb{P}\left[X_{[d]} = x_{[d]}\right]} \cdot \epsilon_{x_{[d]}}\left[X_{[d]}\right] .$$

In **?** the Q-sample has been introduced. It prepares a scheme to realize property 1 (purity) + 2 (q-sampling) of a qpdf, but fails to realize property 3 (q-stochasticity). The q-sample can be prepared for Bayesian Networks, where each child qubit is prepared densely by C-NOTs conditioning on parent qubits.

Q-samples can be prepared by activation circuits acting on uniform quantum states (Hadamard gates acting on ground state).

Doing rejection sampling on the ancilla qubit corresponds with sampling from the normalized contraction with the activation tensor.

Lemma 6. *Given a distribution $\mathbb{P}\left[X_{[d]}\right]$, we construct a circuit preparing its q-sample and add the ancilla encoding of a tensor $\tau\left[X_{[d]}\right]$. The rejection sampling scheme, measuring the ancilla qubit and the $X_{[d]}$ qubits, rejecting the ancilla qubit measured as 0, prepares samples from the distribution*

$$\left\langle \mathbb{P}\left[X_{[d]}\right], \tau\left[X_{[d]}\right]\right\rangle\left[X_{[d]}|\varnothing\right] .$$

3.2 Ancilla Augmentation

For more flexible sampling schemes of Computation-Activation Networks we need to introduce ancilla qubits.

Definition 6 (Ancilla Augmented Distribution)**.** *Let $\mathbb{P}\left[X_{[d]}\right]$ be a probability distribution over variables $X_{[d]}$. Another joint distribution $\tilde{\mathbb{P}}$ of $X_{[d]}$ and ancilla variables $A_{[p]}$ is called an ancilla augmented distribution, if*

$$\tilde{\mathbb{P}}[X_{[d]}|A_{[p]} = \mathbb{I}\left[[p]\right]] = \mathbb{P}\left[X_{[d]}\right] .$$

Sampling from the distribution can be done by rejection sampling on the ancilla augmented distribution, measuring all variables and rejecting all samples where an ancilla variable is 0.

Given an augmented Q-sample of a distribution, we can prepare samples from the distribution by rejection sampling, measuring all variables $X_{[d]}$ and $A_{[p]}$ and rejecting all samples where an ancilla qubit is measured as 0.

When sampling from probability distributions, we can use these samples to estimate probabilistic queries. Building on such particle-based inference schemes, we can perform various inference schemes for Computation-Activation Networks, such as backward inference and message passing schemes.

Given a distribution $\mathbb{P}\left[X_{[d]}\right]$ we add an ancilla variable A and define the augmented distribution (see Figure 5)

$$\tilde{\mathbb{P}}[A, X_{[d]}] = \frac{1}{\prod_{k\in[d]} m_k} \sum_{x_{[d]}\in\bigtimes_{k\in[d]}[m_k]} \epsilon_{x_{[d]}}\left[X_{[d]}\right] \otimes \Big(\mathbb{P}\left[x_{[d]}\right] \cdot \epsilon_1\left[A\right] + (1 - \mathbb{P}\left[x_{[d]}\right]) \cdot \epsilon_0\left[A\right]\Big) .$$

Then we have

$$\tilde{\mathbb{P}}[X_{[d]}|A = 1] = \mathbb{P}\left[X_{[d]}\right] .$$

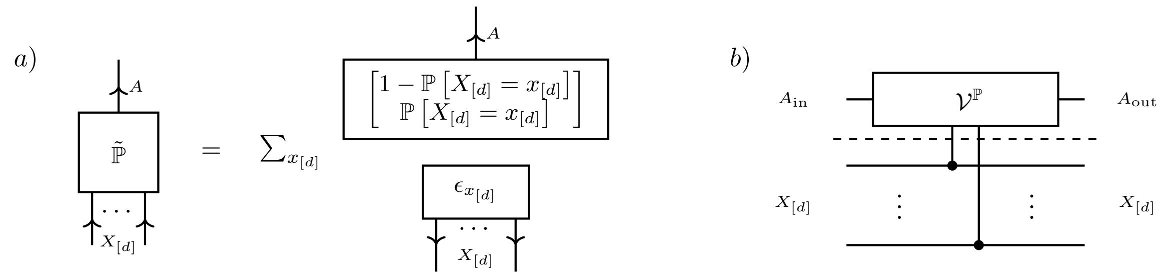


Figure 5: Ancilla augmentation of a distribution $\mathbb{P}\left[X_{[d]}\right]$. a) Augmented distribution $\tilde{\mathbb{P}}[A, X_{[d]}]$ with the property that $\mathbb{P}\left[X_{[d]}\right] = \tilde{\mathbb{P}}[X_{[d]}|A = 1]$. b) Preparation of the augmented distribution by the activation circuit of $\mathbb{P}\left[X_{[d]}\right]$.

3.3 Amplitude Amplification

Note, that the variable qubits are uniformly distributed when only the computation circuit is applied. When sampling the probability distribution, we need the aniclla qubits to be in state 1 in order for the sample to be valid. Any other states will have to be rejected.

Classically, this can be simulated in the same way: Just draw the variables from uniform, calculate the value qubit by a logical circuit inference and accept with probability by the computed value.

For this procedure to be more effective (and in particular not having an efficient classical pendant), we need amplitude amplification on the value qubit. This can provide a square root speedup in the complexity compared with classical rejection sampling.

Open Question: Is there a way to avoid amplitude amplification and use a more direct circuit implementation of the activation network? - Cannot be the case, when the encoding is determined by the activation tensor alone: Needs to use the computated statistic as well.

3.4 Sampling from Computation-Activation Networks as Quantum Circuits

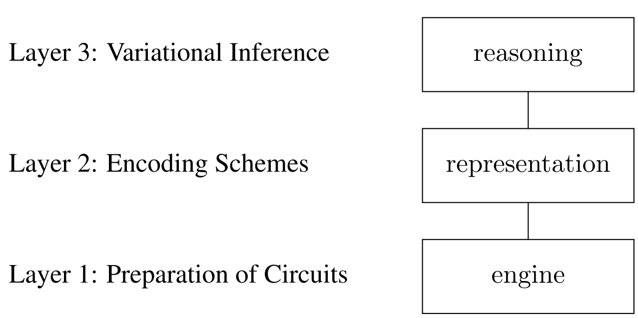
So far: Sample from Hybrid Logic Networks, would need qudits for for more general Computation-Activation Networks.

tnreason provides tensor network representations of knowledge bases and exponential families following a Computation Activation architecture. Here are some ideas to utilize quantum circuits for sampling from Computation-Activation Networks. We can produce Q-samples for ancilla augmented Computation-Activation Networks using computation circuits and activation circuits:

- For each (sub-) statistic, prepare a qubit by Computation circuits
- Based on the computed qubits, prepare ancilla qubits by Activation circuits to the activation cores.

4 Implementation

The introduced quantum circuit preparation schemes have been implemented in the python package qcreason, which consists in three layers:



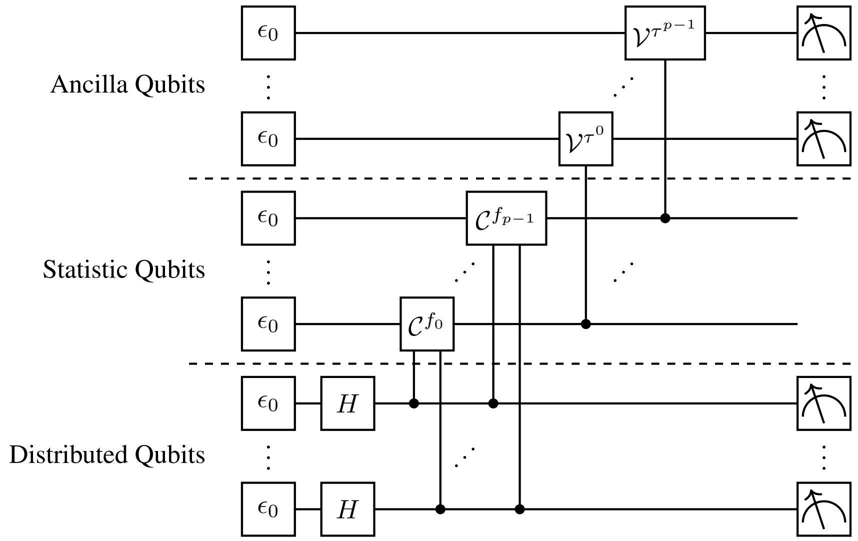


Figure 6: Quantum Circuit to reproduce a Computation-Activation Network (with elementary activation) by rejection sampling. We measure the distributed qubits $X_{[d]}$ and the ancilla qubits $A_{[p]}$ and reject all samples, where an ancilla qubit is measured as 0.

A Extension: Sampling from proposal distributions

We can prepare basis circuit encodings to selection augmented formulas, in this way introducing formula selecting networks.

Idea for an inductive reasoning scheme: Prepare a q-sample from the empirical distribution and the current distribution. Then prepare the basis circuit encodings, where the selection variables are shared and the distributed variables assigned to the prepared samples. Now, the ancilla qubits can be designed to ϵ_1 and ϵ_0 accordingly. The rejection sampling scheme on both ancillas being 1 and the measurement of L prepares then the distribution

$$\left\langle \left\langle \mathbb{P}^D \left[X_{[d]}\right], \sigma^{\mathcal{F}} \left[X_{[d]}, L\right] \right\rangle [L], \left\langle \tilde{\mathbb{P}} \left[X_{[d]}\right] \right\rangle [L] \right\rangle [L|\varnothing]$$

That is, the probability of selecting l is proportional to

$$\mu_D [L = l] \cdot (1 - \tilde{\mu} [L = l])$$

and thus prefers formulas, which have a large empirical mean, but a small current mean.

Open Question: Since the distribution is "similar" to $\exp [\mu_D [L = l] - \tilde{\mu} [L = l])]$ (terms appear in Taylor of first order), can we tune the distribution with an inverse temperature parameter β?

B Comparing tensor networks and quantum circuits

First of all, we need to extend to complex tensors, which are maps

$$\tau : \bigtimes_{k \in [d]} [2] \to \mathbb{C}$$

with image in $\mathbb{C}$ instead of $\mathbb{R}$ as in the report.

A coarse comparison of the nomenclature used for quantum circuits and tensor networks:

Quantum Circuit	**Tensor Network**
Qubit	Boolean Variable
Quantum Gate	Unitary Tensor
Quantum Circuit	Tensor Network on a graph

Some constraints appear for a tensor network to be a quantum circuit

- **Unitarity of each gate:** That is the variables of each tensor are bipartite into sets A^{in} and A^{out} of same cardinality and the basis encoding with respect to this bipartition, that is

$$T_{\mathrm{in}\to\mathrm{out}}[X_{\mathrm{in}}, X_{\mathrm{out}}] : \bigotimes_{k\in A^{\mathrm{in}}} \mathbb{C}^2 \to \bigotimes_{k\in A^{\mathrm{out}}} \mathbb{C}^2 \,,$$

is a unitary map, that is

$$(T_{\mathrm{in}\to\mathrm{out}})^H \circ (T_{\mathrm{in}\to\mathrm{out}}) = \left\langle T_{\mathrm{in}\to\mathrm{out}}[X_{\mathrm{in}}, Y], \overline{T}_{\mathrm{in}\to\mathrm{out}}[Y, X_{\mathrm{out}}] \right\rangle [X_{\mathrm{out}}, X_{\mathrm{in}}] = \delta\left[X_{\mathrm{out}}, X_{\mathrm{in}}\right].$$

- **Incoming-Outgoing structure:** Variable appear at most once as incoming and at most once as outgoing variables. Those not appearing as outgoing (respectively as incoming) are the input and the output variables of the whole circuit.
- **Acyclicity:** Incoming and outgoing variables of each tensor core provide a direction of each edge tensor. With respect to this directionality the graph underlying the tensor network has to be acyclic.

The unitary tensors can be aligned layerwise, if and only if the last two assumption hold, i.e. the directed graph is acyclic and each variable appears at most once as an incoming and at most once as an outgoing variable.

C POVM measurements as contractions

The main difficulty of using quantum circuits as contraction providers is that we can only extract information through measurements. Therefore measurement is the only way to execute contractions of the circuit, which come with restrictions when interested in contraction with open variables.

The most general measurement formalism is through a POVM, a set $\{E_y \,:\, y \in [r]\}$ of positive operators with

$$\sum_{y\in[r]} E_y = I$$

Measuring a pure state $|\psi\rangle$ We then get outcome m with probability

$$\langle\psi|E_y|\psi\rangle$$

We define a measurement variable Y taking indices $y \in [r]$ and a measurement tensor

$$E[Y, X_{\mathrm{in}}, X_{\mathrm{out}}]$$

with slices

$$E[Y = y, X_{\mathrm{in}}, X_{\mathrm{out}}] = E_y \,.$$

Repeating the measurement asymptotically on a state $|\psi\rangle$ prepared by a quantum circuit $\tau^{\mathcal{G}}$ acting on the trivial start state $\mathbb{I}$, we denote the measurement outcome by y^j. In the limit $m \to \infty$ we get almost surely

$$\frac{1}{m}\sum_{j\in[m]} \epsilon_{y^j}\left[Y\right] \to \left\langle \tau^{\mathcal{G}}[X_{\mathrm{in}}], E[Y, X_{\mathrm{in}}, X_{\mathrm{out}}], \tau^{\tilde{\mathcal{G}}}[X_{\mathrm{out}}] \right\rangle [Y] \,.$$

POVMs to computational basis measurements of subsets of qubits are constructed as products with delta tensors on the non-measured qubits.

D Alternative Contraction Provider: Overlap-measuring Circuits

Quantum Circuits such as the SWAP test and the Hadamard test can be used to measure overlaps of quantum states, which are the squared absolutes of contractions of two state tensors.

- When we have preparation schemes for two tensors, we can control them with a common ancilla qubit and measure their contraction by a Hadamard test (alternatively, using the SWAP test and state preparation in two registers).
- Can we extend these schemes to contractions of more general tensor networks?