VERIFICATION OF QUANTUM CIRCUITS

Lecture 4: An automata-based framework for Verification & Bug hunting in Quantum circuits - Part II

Reference: Paper with the same name as above
- by Chen, Chung, Lengál, Lin, Tsai and Yen (PLDI '23)

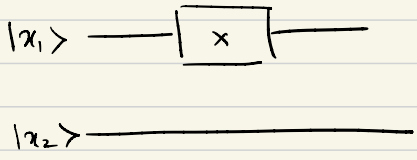
Plan for the lecture:

- 1. Quantum gates as symbolic tree updates
- 2. Tree automata for the update operation
- 3. Putting things together

Part 1 : Quantum gates as tree updates.

We illustrate the key ideas using an example.

Consider the following gate:



$$X : \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

↓ quantum version of NOT gate

If there is a single qubit $|x_1\rangle = c_0|0\rangle + c_1|1\rangle$

$$X \cdot \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_0 \end{bmatrix}$$

↳ flips the numbers

Now we have a 2 qubit system.

$$c_{00} \cdot |00\rangle + c_{01} \cdot |01\rangle + c_{10} \cdot |10\rangle + c_{11} \cdot |11\rangle$$

The action of the gate can be represented by the following matrix:

$$X' = X \otimes \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{I_1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

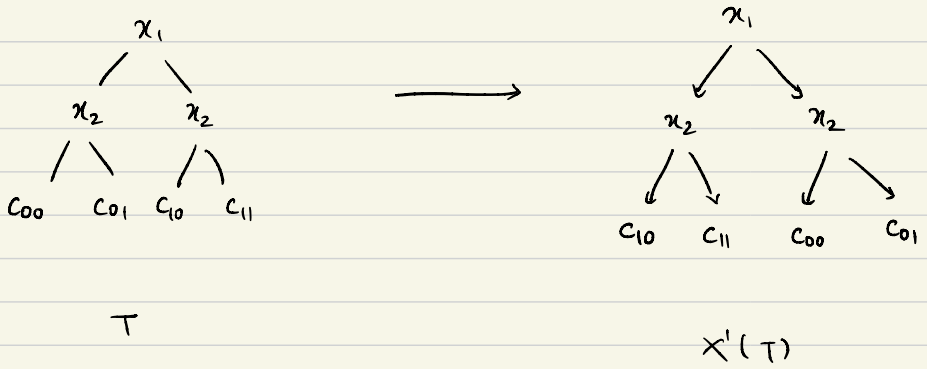
$$X' \cdot \begin{bmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{bmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{pmatrix} = \begin{pmatrix} c_{10} \\ c_{11} \\ c_{00} \\ c_{01} \end{pmatrix}$$

↳ swaps amplitudes of states where x_1's value is 1 with states where x_1's value is 0.

$\therefore$ If $T = c_{00} \cdot |00\rangle + c_{01} \cdot |01\rangle + c_{10} \cdot |10\rangle + c_{11} \cdot |11\rangle$

Then $X'(T) = c_{10} \cdot |00\rangle + c_{11} \cdot |01\rangle + c_{00} \cdot |10\rangle + c_{01} \cdot |11\rangle$

Viewed as a tree:



T $\qquad$ $X'(T)$

The subtrees below x_1 are swapped by this operation.

Key idea: Define symbolic tree-update operations that capture the effect of a circuit.

Projection & Restriction:

Consider the tree as a function: $T : \{0,1\}^n \longrightarrow \mathbb{Z}^5$

also represents a quantum state.

Projection:

$$T_{x_t}(b_1 b_2 \ldots b_t \ldots b_n) = T(b_1 b_2 \ldots 1 \ldots b_n)$$

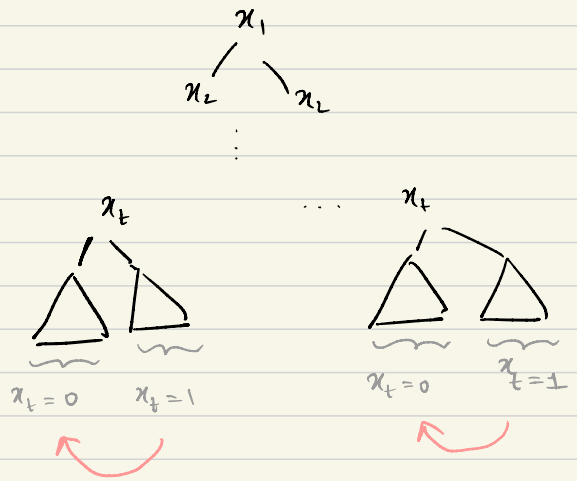
$$T_{\overline{x_t}}(b_1 b_2 \ldots b_t \ldots b_n) = T(b_1 b_2 \ldots 0 \ldots b_n)$$

Explanation of the projection operation comes in the next page.

$$T_{x_t}(b_1 b_2 \dots b_t \dots b_n) = T(b_1 b_2 \dots 1 \dots b_n)$$

$$T_{\bar{x}_t}(b_1 b_2 \dots b_t \dots b_n) = T(b_1 b_2 \dots 0 \dots b_n)$$

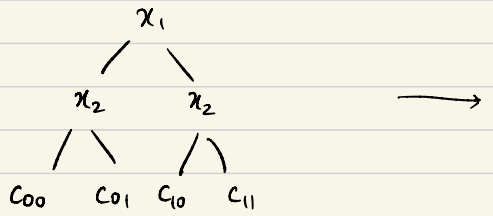
T_{x_t} :



: T_{x_t} copies right subtree of x_t to its left

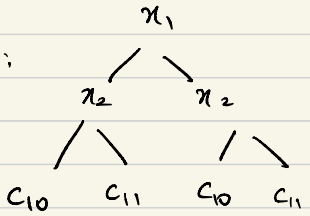
Example:

T :



$\longrightarrow$

T_{x_1} :



Similarly, $T_{\bar{x}_t}$ copies left subtree of x_t to its right.

Seen as a state: if $T = c_{00} \cdot |00\rangle + c_{01} \cdot |01\rangle + c_{10} \cdot |10\rangle + c_{11} \cdot |11\rangle$

$$T_{x_1} = c_{10} \cdot |00\rangle + c_{11} \cdot |01\rangle + c_{10} \cdot |10\rangle + c_{11} \cdot |11\rangle$$

Restriction:

$$B_{x_t}(b_1 b_2 \dots b_t \dots b_n) = b_t \quad \rightarrow \text{simply returns bit } b_t$$

$$B_{\bar{x}_t}(b_1 b_2 \dots b_t \dots b_n) = \bar{b_t} \quad \rightarrow \text{returns the complement of } b_t$$

Now, suppose $T = c_{00} \cdot |00\rangle + c_{01} \cdot |01\rangle + c_{10} \cdot |10\rangle + c_{11} \cdot |11\rangle$

$$T_{x_1} = c_{10} \cdot |00\rangle + c_{10} \cdot |01\rangle + c_{10} \cdot |10\rangle + c_{11} \cdot |11\rangle$$

$$B_{\bar{x}_1} \cdot T_{x_1} = c_{10} \cdot |00\rangle + c_{10} \cdot |01\rangle$$

Revisiting the X' circuit:

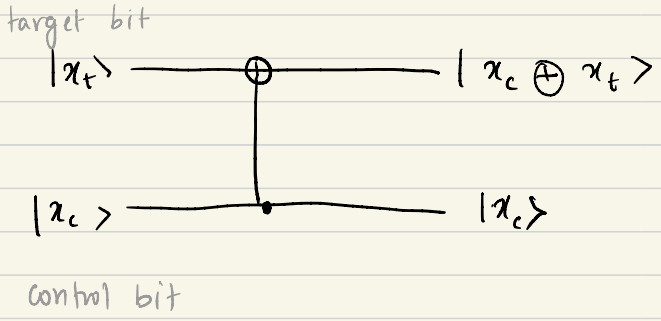
$$X'(T) = c_{10} \cdot |00\rangle + c_{10} \cdot |01\rangle + c_{10} |10\rangle + c_{11} \cdot |11\rangle$$

Notice that:

$$X'(T) = B_{\bar{x}_1} \cdot T_{x_1} + B_{x_1} \cdot T_{\bar{x}_1}$$

Therefore we have written the effect of the circuit in terms of the two update operations

Example: The CNOT gate (controlled NOT gate)



$$\text{CNOT}: \quad \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

bit 1 is x_c
bit 2 is x_t

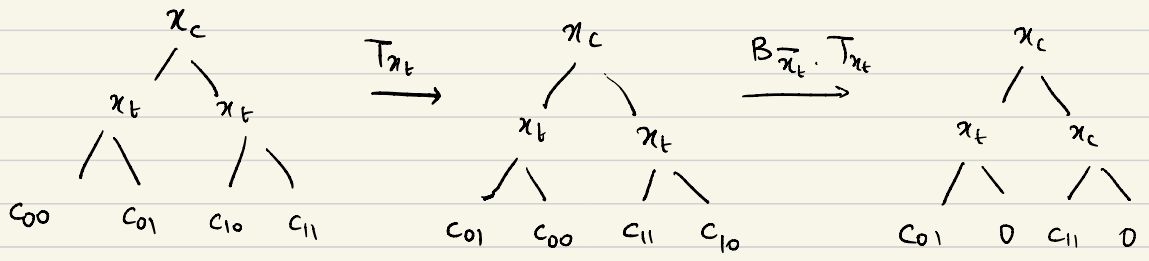
If $x_c = 1$ then it flips x_t

Else maintains x_t as it is

$$\text{CNOT} \cdot \begin{bmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{bmatrix} = \begin{bmatrix} c_{00} \\ c_{01} \\ c_{11} \\ c_{10} \end{bmatrix}$$

The tree update formula for CNOT is as follows:

$$\text{CNOT}_t^c(T) = B_{\overline{x_c}} \cdot T + B_{x_c} \cdot \left(B_{\overline{x_t}} \cdot T_{x_t} + B_{x_t} \cdot T_{\overline{x_t}} \right)$$



Interpretation of the formula:

$$CNOT_t^c(T) = B_{\bar{x}_c} \cdot T + B_{x_c} \cdot \left(B_{\bar{x}_t} \cdot T_{x_t} + B_{x_t} \cdot T_{\bar{x}_t} \right)$$

If $x_c = 0$ then

target remains same

If $x_c = 1$

apply X gate to target bit

Part 2: Tree automata to capture the tree update operations

Example: The X_t gate

Assume x_t stands for the target qubit on which the X gate is applied.

The tree update characterization is as follows:

$$X_t\ (T) \ := \ B_{\bar{x}_t} \cdot T_{x_t} + B_{x_t} \cdot T_{\bar{x}_t}$$

↳ swap the left and right subtrees of x_t.

→ Suppose $\mathcal{A}$ is a tree automaton representing an input set of quantum states

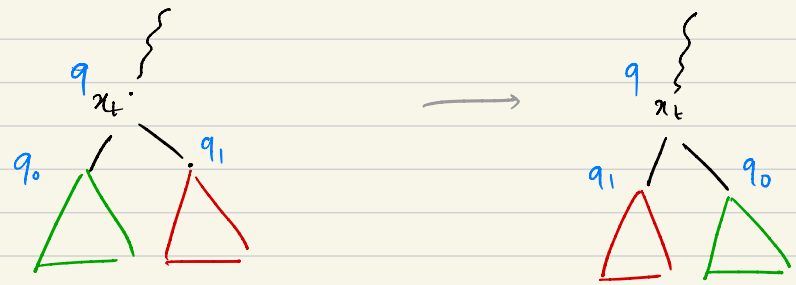
Let $X_t(\mathcal{A})$ (→ represents a tree automaton) be obtained as follows:

every: $q \xrightarrow{x_t} (q_0, q_1)$ ⟶ $q \xrightarrow{x_t} (q_1, q_0)$

in $\mathcal{A}$ in $\mathcal{A}_t$

- all other transitions remain the same.

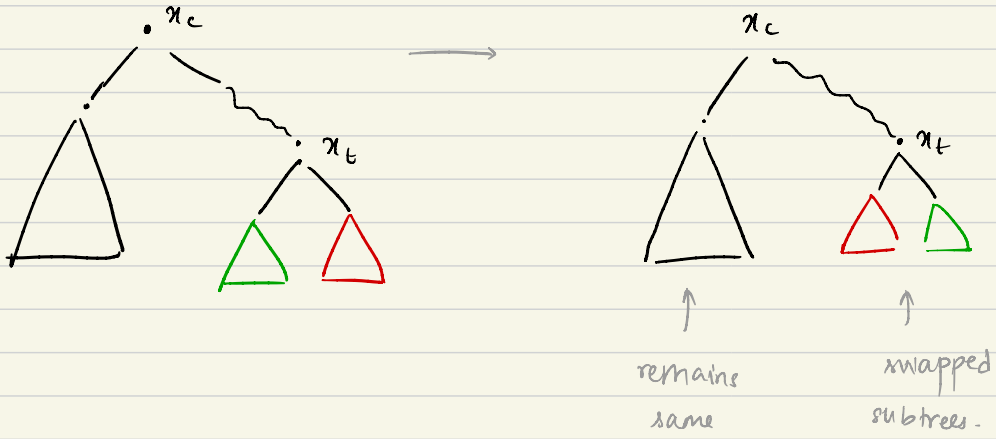
Essentially:



Example: The $CNOT_t^c$ gate

$$CNOT_t^c(T) = B_{\overline{x_c}} \cdot T + B_{x_c} \cdot (B_{\overline{x_t}} \cdot T_{x_t} + B_{x_t} \cdot T_{\overline{x_t}})$$

Assume $c < t$. Then:



Tree automaton for $CNOT_t^c$:

→ Suppose $\mathcal{A}$ is a tree automaton representing an input set of states

→ The automaton for $CNOT_t^c$ is constructed in the following steps:

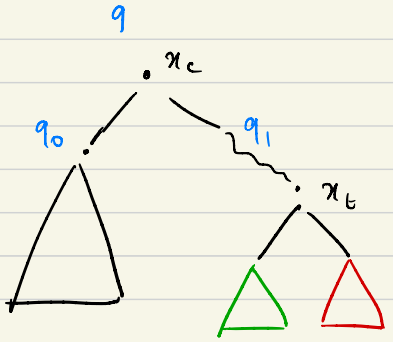
- **Step 1:** $\mathcal{A}_1 = X_t(\mathcal{A})$ [the automaton constructed in previous example]

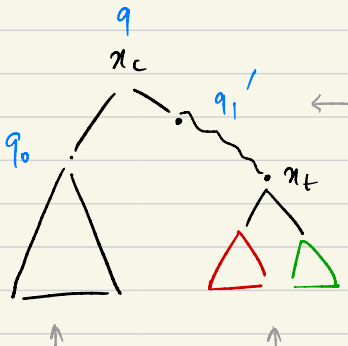
- **Step 2:** $\mathcal{A}_1' = (Q_1', \Sigma, \Delta_1', R')$
  ↳ obtained from $\mathcal{A}_1$ by priming all occurrences of states.

- **Step 3:**
  $\mathcal{A}^R := \langle Q \cup Q_1', \Sigma, \Delta^R \cup \Delta_1', R \rangle$ where

$$\Delta^R = \{ q \xrightarrow{x_k} (q_0, q_1) \mid q \xrightarrow{x_k} (q_0, q_1) \in \Delta \wedge k \neq c \}$$
$$\cup \{ q \xrightarrow{x_c} (q_0, q_1') \mid q \xrightarrow{x_c} (q_0, q_1) \in \Delta \} \cup \{ t \mid t \in \Delta_e \}$$

leaf transitions



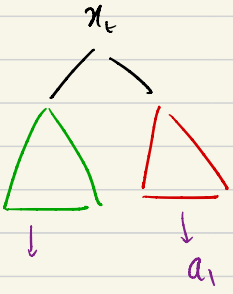


primed copy implements X_t gate → so it swaps left and right subtree of x_t.

remains same

swapped subtrees.

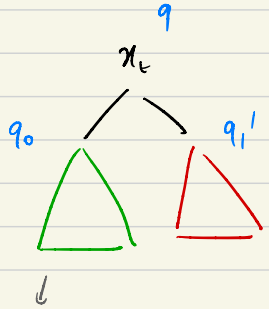
Example: Consider a gate $U(T) = a_1 \cdot B_{x_t} \cdot T + a_0 \cdot B_{\overline{x_t}} \cdot T$



scale by a_0

→ $U(T)$ scales left subtree of x_t by a_0 and right subtree by a_1.

Idea:



→ a primed version where all the leaf transitions are multiplied by a_1

all leaf transitions of original automaton are multiplied by a_0.

full construction:

→ $\mathcal{A} = (Q, \Sigma, \Delta, R)$ the input tree automaton

→ $\mathcal{A}_1' = \langle Q', \Sigma, \Delta_1', R' \rangle$ where

$$\Delta_1' = \Delta_i' \cup \{ q' \xrightarrow{a_1 \cdot c} () \mid q \xrightarrow{c} () \in \Delta_e \}$$

Δ_i': primed version of internal transitions from Δ

- $\mathcal{A}^R = \langle Q \cup Q', \Sigma, \Delta^R \cup \Delta_1', R \rangle$ where

$$\Delta^R = \{ q \xrightarrow{a_0 \cdot c} () \mid q \xrightarrow{c} () \in \Delta_e \}$$

$$\cup \{ q \xrightarrow{x_k} (q_0, q_1) \mid q \xrightarrow{x_t} (q_0, q_1) \in \Delta_e \wedge k \neq t \}$$

$$\cup \{ q \xrightarrow{x_k} (q_0, q_1') \mid q \xrightarrow{x_k} (q_0, q_1) \in \Delta_e \text{ and } k = t \}$$

Part 3: Putting things together

- In the paper they provide formulas for many quantum gates including the universal one Toffoli

Compositional approach for general formulas:

- The formulas are projection, restriction, adding/subtracting terms, and multiplying by scalars 'ω^2' and $\frac{1}{\sqrt{2}}$.
- Given a tree automaton representing input states, they give a tree automaton representing the application of projection and restriction.

For example: $X'(T) = B_{\bar{x}_1} \cdot T_{x_1} + B_{x_1} \cdot T_{\bar{x}_1}$

Given TA A_p for input states

→ construct TA for $\{ B_{\bar{x}_1} \cdot T_{x_1}(T) \mid T \in \mathcal{L}(A_p) \}$

→ construct TA for $\{ B_{x_1} \cdot T_{\bar{x}_1}(T) \mid T \in \mathcal{L}(A_p) \}$

→ Then apply construction for the $+$.

- Automata for $\pm$ and multiplication by ω^2 are also presented.

↳ refer to Sections 5 & 6 of the paper for further details.

Summary:

→ Quantum state as trees

→ Set of quantum states as a tree automaton

→ Quantum gates as "symbolic update" operations, whose effect can be emulated using tree automata

→ Using a compositional approach, the tree automaton representing the effect of the full circuit is computed.

→ Paper presents non-trivial and encouraging experimental results

↳ cases when existing circuit equivalence checkers failed, the current method produced bugs

→ Exciting link between quantum circuits & automata theory. More left to be explored.