VERIFICATION OF QUANTUM CIRCUITS

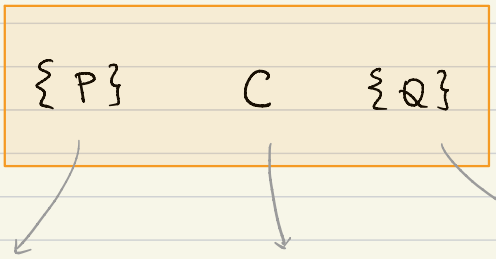
Lecture 3: An automata-based framework for Verification & Bug hunting in Quantum circuits

Reference: Paper with the same name as above
- by Chen, Chung, Lengál, Lin, Tsai and Yen (CACM'25)

Plan for the lecture:

1. The verification question; its applications; and an idea of its solution
2. Notations for quantum states & gates
3. Tree automata
4. Sets of quantum states as languages of tree automata
5. Quantum gates as tree updates } next lecture
6. Solving the verification question

Part 1: The verification question and its applications



Set of input quantum state — Pre-condition

a Quantum circuit

Set of output quantum states — Post-condition

Question: Given the set P of quantum states as input to circuit C, is the output included in Q ?

Idea for the solution:

1. Represent {P} as the language of a tree automaton $\mathcal{A}_P$
   - each quantum state is encoded as a finite tree.
2. The operation of C on an input quantum state is modeled as an update to the tree representing the state.
3. Given $\mathcal{A}_P$ and C there is an algorithm that outputs a tree automaton $\mathcal{A}_P'$: this represents the set of output states that can be obtained on applying C to P.
4. Check if $\mathcal{A}_P' \subseteq \mathcal{A}_Q$
   - Tree automaton representing the set of quantum states Q.

Application to Quantum circuit equivalence:

A typical challenge in hardware design:

Circuit C

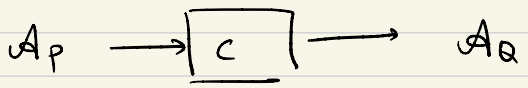
↓

Optimized circuit C'

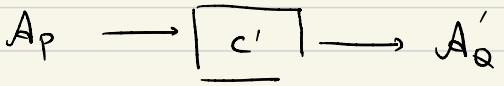
Are C, C' equivalent?

Using $\{P\}\ C\ \{Q\}$ to under-approximate the equivalence test:

- Pick a set P (created by user or automatically)
- Construct tree automaton $\mathcal{A}_Q$ representing the output of applying C to P.



Similarly construct $\mathcal{A}'_Q$:



If $\mathcal{L}(\mathcal{A}_Q) \neq \mathcal{L}(\mathcal{A}'_Q)$ then there is a bug: C' is not equivalent to C

Else, we can say that both circuits behave the same on P.

→ Try multiple P's to catch bugs. This is a bug-hunting procedure & not a complete equivalence test.

→ See the paper for more applications & experiments & context.

Part 2: Some preliminaries on quantum gates & circuits.

→ In a quantum system with n qubits

→ basis states are $\{ |j\rangle \mid j \in \{0,1\}^n \}$

→ a quantum state is a superposition of all the basis states

$$\sum_{j \in \{0,1\}^n} a_j |j\rangle$$

where $a_j \in \mathbb{C}$ and $\sum_{j \in \{0,1\}^n} |a_j|^2 = 1$

function representation of a quantum state:

$$T : \{0,1\}^n \longrightarrow \mathbb{C}$$ where $T(j) = a_j$ for $j \in \{0,1\}^n$

Example:

Quantum state $\frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$

$$= \left[\frac{1}{\sqrt{2}}, 0, 0, \frac{1}{\sqrt{2}} \right]^T$$

$$= \text{function } T = \left\{ 00 \mapsto \frac{1}{\sqrt{2}}, 01 \mapsto 0, 10 \mapsto 0, 11 \mapsto \frac{1}{\sqrt{2}} \right\}$$

Representing complex numbers:

→ we consider a subset of complex numbers that can be expressed by the following algebraic encoding:

$$\left(\frac{1}{\sqrt{2}} \right)^k \left(a + b\omega + c\omega^2 + d\omega^3 \right)$$

where $a, b, c, d, k \in \mathbb{Z}$, $\omega = e^{i\pi/4}$ (unit vector that makes an angle of 45° with +ive real axis)

→ A complex number is then represented as a 5 tuple:

$$(a, b, c, d, k)$$

Remark: Notice that the subset of complex numbers that can be represented this way is countable, whereas $\mathbb{C}$ is uncountable

However, the paper states that this subset is sufficient to describe a set of quantum gates that can implement universal quantum computation.

→ With such an encoding, a quantum state can be represented as:

$$T : \{0,1\}^n \longrightarrow \mathbb{Z}^5$$

— Rest of the concepts about quantum gates and circuits are as discussed in the last lecture.

Part 3: Tree automata

→ We consider finite complete binary trees.

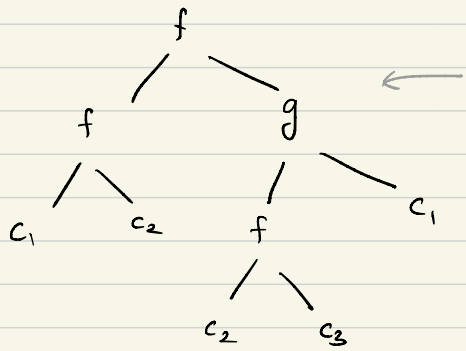
Alphabet: Σ is a "ranked alphabet" consisting of two kinds of symbols:

binary symbols: $f, g, \ldots$

constant symbols: $c_1, c_2, \ldots$

Binary tree:

$T =$



Internal nodes are binary symbols

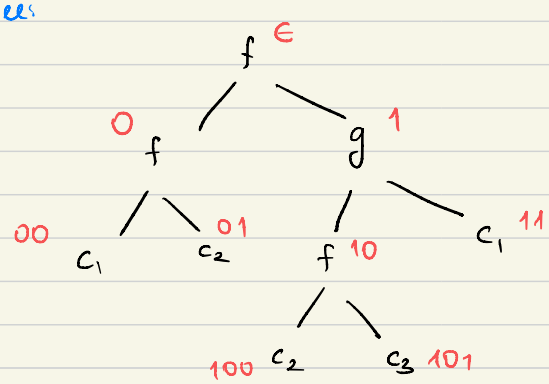
leaves are constants.

→ Internal nodes have two branches.

Tree T above can also be written as a term:

$$f\ (\ f(c_1, c_2)\ ,\ g\ (\ f(c_2, c_3)\ ,\ c_1)\ )$$

Nodes of a binary tree:



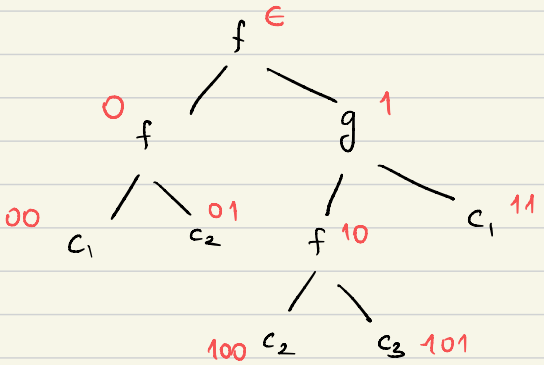
→ Root is ϵ
→ for a node 'a' its left child is a0 right child is a1

Labeling function:

- Call the set of nodes of a tree to be N_T
- The labeling function L_T maps N_T to Σ

For example:

T:



$N_T = \{ \epsilon, 0, 1, 00, 01, 10, 11, 101, 100 \}$

$L_T = \{ \epsilon \mapsto f$
$0 \mapsto f$
$1 \mapsto g$
$00 \mapsto c_1$
$01 \mapsto c_2$
$10 \mapsto f$
$11 \mapsto c_1$
$100 \mapsto c_2$
$101 \mapsto c_3 \}$

Tree automata:

an acceptor of a tree language

Example 1: $\Sigma = \{ f, g, c \}$

f, g: binary; c: constant

$L_1 = \{$ balanced trees of height 2 that contain 'f' $\}$

each path contains 2 internal nodes

$q_0 \xrightarrow{f} (p_1, p_1)$ $\quad p_1 \xrightarrow{f,g} (p_2, p_2)$ $\quad p_2 \xrightarrow{c} ()$

$q_0 \xrightarrow{g} (q_1, p_1)$ $\quad q_1 \xrightarrow{f} (p_2, p_2)$

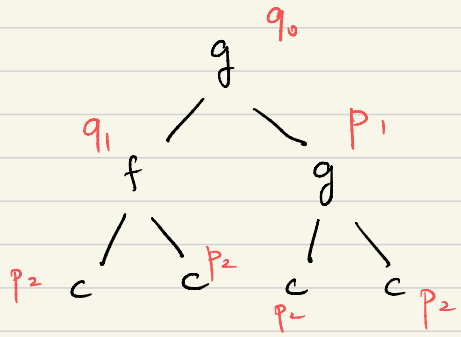
$q_0 \xrightarrow{g} (p_1, q_1)$

since first node is not 'f', the transitions guess where f would appear.

Top transition guesses 'f' appears on the left branch

Bottom transition guesses 'f' appears on the right branch.

Example of a run:



→ A tree where 'f' does not appear will not have a run at all in this automaton.

Example 2:

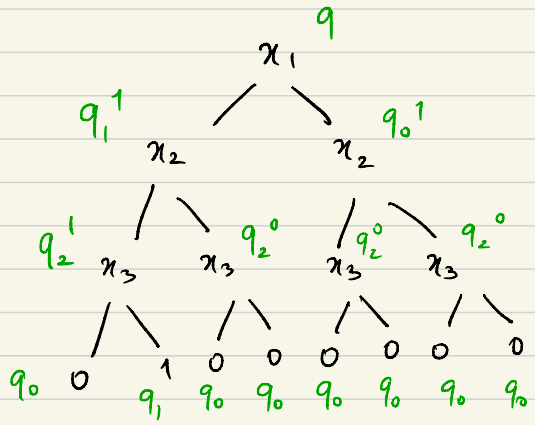
$$\Sigma = \{ x_1, x_2, x_3, \quad 0, 1 \}$$

binary symbols — constant symbols

Consider the automaton A_3 given as follows:

$q \xrightarrow{x_1} (q_0^1, q_1^1)$	$q_1^1 \xrightarrow{x_2} (q_0^2, q_1^2)$	$q_1^2 \xrightarrow{x_3} (q_0, q_1)$	$q_0 \xrightarrow{0} ()$
$q \xrightarrow{x_1} (q_1^1, q_0^1)$	$q_1^1 \xrightarrow{x_2} (q_1^2, q_0^2)$	$q_1^2 \xrightarrow{x_3} (q_1, q_0)$	$q_1 \xrightarrow{1} ()$
	$q_0^1 \xrightarrow{x_2} (q_0^2, q_0^2)$	$q_0^2 \xrightarrow{x_3} (q_0, q_0)$	

↳ accepts balanced trees of height 3 with exactly one leaf being 1.



In the above tree, say T, node 001 is labeled 1, every other leaf is 0.

$$\therefore T(001) = 1$$

$$T(j) = 0 \quad \forall j \in \{0,1\}^3 \setminus \{001\}$$

formal definition:

Non-deterministic Tree Automaton:

$$\mathcal{A} = (Q, \Sigma, \Delta, R)$$

- Q: finite set of states
- Σ: finite alphabet
- Δ: transitions
- R: root states (initial states)

$$\Delta : \Delta_i \cup \Delta_\ell$$

- Δ_i: internal transitions
- Δ_ℓ: leaf transitions

$$q \xrightarrow{x} (q_1, q_2) \qquad q \xrightarrow{c} ()$$

→ No separate acceptance condition. Any tree that has a run is accepted.

Part 4: Encoding sets of quantum states using tree automata.

We explain this encoding using examples.

Example 1: All basic states of a 2-qubit system:

$|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$

$|00\rangle$:

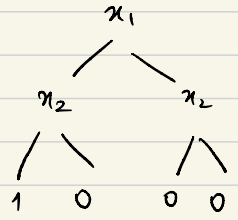
$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

as a vector

T_{00} :

$00 \mapsto 1$
$01 \mapsto 0$
$10 \mapsto 0$
$11 \mapsto 0$

as a function



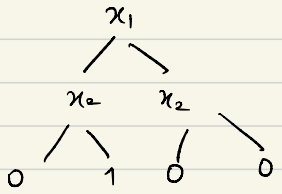
as a tree

$|01\rangle$:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

T_{01} :

$00 \mapsto 0$
$01 \mapsto 1$
$10 \mapsto 0$
$11 \mapsto 0$

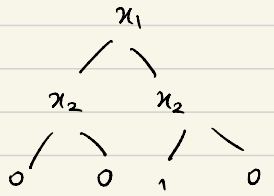


$|10\rangle$:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

T_{10} :

$00 \mapsto 0$
$01 \mapsto 0$
$10 \mapsto 1$
$11 \mapsto 0$

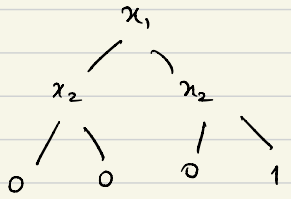


$|11\rangle$:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

T_{11} :

$00 \mapsto 0$
$01 \mapsto 0$
$10 \mapsto 0$
$11 \mapsto 1$



Tree automaton $\mathcal{A}_2$ representing the tree representation of $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$:

$q \xrightarrow{x_1} (q_0^1, q_1^1)$

$q \xrightarrow{x_1} (q_1^1, q_0^1)$

$q_0^1 \xrightarrow{x_2} (q_0, q_0)$

$q_1^1 \xrightarrow{x_2} (q_1, q_0)$

$q_1^1 \xrightarrow{x_2} (q_0, q_1)$

$q_0 \xrightarrow{0} ()$

$q_1 \xrightarrow{1} ()$

This idea can be generalized to any 'n'.

↳ Part 3 - Example 2 gave the automaton $\mathcal{A}_3$ for $n=3$.

→ We can construct automaton $\mathcal{A}_n$: representing all basic states of an n-qubit system.

Example 2: Consider the finite set of states:

$$P = \left\{ \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \,,\; \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle \right\}$$

$$\frac{1}{\sqrt{2}} = \left(\frac{1}{\sqrt{2}}\right)^1 (1 + 0.\omega + 0.\omega^2 + 0.\omega^3)$$

$\therefore$ Encoding of $\frac{1}{\sqrt{2}} = (1, 0, 0, 0, 1)$

Automaton $\mathcal{A}_P$: two initial states $q_{00,11}$ and $q_{01,10}$

$$q_{00,11} \xrightarrow{x_1} (q_{00}, q_{11}) \qquad q_{00} \xrightarrow{x_2} (q_c, q_0)$$

$$q_{11} \xrightarrow{x_2} (q_0, q_c)$$

$$q_{01,10} \xrightarrow{x_1} (q_{01}, q_{10}) \qquad q_{01} \xrightarrow{x_2} (q_0, q_c)$$

$$q_{10} \xrightarrow{x_2} (q_c, q_0)$$

$$q_0 \xrightarrow{0} ()$$

0 stands for $(0,0,0,0,0)$

$$q_c \xrightarrow{(1,0,0,0,1)} ()$$

Remark: In the paper, they consider only finite sets of states, and hence finite tree languages.