Theory of Computation
Tree and Quantum Automata - A Brief Introduction
Why Tree Automata?

- Foundations of XML type languages (DTD, XML Schema, Relax NG...)
- Provide a general framework for XML type languages
- A tool to define regular tree languages with an operational semantics
- Provide algorithms for efficient validation
- Basic tool for static analysis (proofs, decision procedures in logic)
- ...

E.g. Binary trees with an even number of $a$'s
Binary Trees & Ranked Trees

- Binary trees with an even number of a’s
- How to write transitions?
  - (even, odd) $\xrightarrow{a}$ even
  - (even, even) $\xrightarrow{a}$ odd
  - ...

- Ranked Tree:
  - Alphabet:
    \{a^{(2)}, b^{(2)}, c^{(3)}, #^{(0)}\}
  - $a^{(k)}$: symbol $a$ with arity($a$) = $k$
A ranked bottom-up tree automaton $A$ consists of:

- $\text{Alphabet}(A)$: finite alphabet of symbols
- $\text{States}(A)$: finite set of states
- $\text{Rules}(A)$: finite set of transition rules
- $\text{Final}(A)$: finite set of final states ($\subseteq \text{States}(A)$)

where $\text{Rules}(A)$ are of the form $(q_1, \ldots, q_k)^{a(k)} \rightarrow q$; if $k = 0$, we write $\epsilon^{a(0)} \rightarrow q$.
**Bottom-up Tree Automata: An Example**

**Principle**
- **Alphabet** \( A = \{ \land, \lor, 0, 1 \} \)
- **States** \( A = \{ q_0, q_1 \} \)
- 1 accepting state at the root: \( \text{Final}(A) = \{ q_1 \} \)

**Rules**
- \( \epsilon \xrightarrow{0} q_0 \)
- \( \epsilon \xrightarrow{1} q_1 \)
- \( (q_1, q_1) \xrightarrow{\land} q_1 \)
- \( (q_0, q_1) \xrightarrow{\lor} q_1 \)
- \( (q_0, q_1) \xrightarrow{\lor} q_0 \)
- \( (q_1, q_0) \xrightarrow{\lor} q_0 \)
- \( (q_0, q_0) \xrightarrow{\lor} q_0 \)
- \( (q_0, q_0) \xrightarrow{\land} q_0 \)
- \( (q_0, q_0) \xrightarrow{\land} q_0 \)
A ranked top-down tree automaton $A$ consists of:

- $\text{Alphabet}(A)$: finite alphabet of symbols
- $\text{States}(A)$: finite set of states
- $\text{Rules}(A)$: finite set of transition rules
- $\text{Final}(A)$: finite set of final states ($\subseteq \text{States}(A)$)

Where $\text{Rules}(A)$ are of the form $q \xrightarrow{a^{(k)}} (q_1, \ldots, q_k)$; if $k = 0$, we write $\epsilon \xrightarrow{a^{(0)}} q$

Top-down tree automata also recognize all regular tree languages
Top-down Tree Automata: An Example

Principle

- starting from the root, guess correct values
- check at leaves
- 3 states: $q_0, q_1, acc$
- initial state at the root: $q_1$
- accepting if all leaves labeled $acc$

Transitions

$q_1 \xrightarrow{\wedge} (q_1, q_1)$  $q_1 \xrightarrow{\lor} (q_0, q_1)$
$q_0 \xrightarrow{\wedge} (q_0, q_1)$  $q_1 \xrightarrow{\lor} (q_1, q_0)$
$q_0 \xrightarrow{\wedge} (q_1, q_0)$  $q_1 \xrightarrow{\lor} (q_1, q_1)$
$q_0 \xrightarrow{\wedge} (q_0, q_0)$  $q_0 \xrightarrow{0} (q_0, q_0)$
$q_1 \xrightarrow{1} acc$  $q_0 \xrightarrow{0} acc$
Theorem 1

The following properties are equivalent for a tree language $L$:

(a) $L$ is recognized by a **bottom-up non-deterministic** tree automaton
(b) $L$ is recognized by a **bottom-up deterministic** tree automaton
(c) $L$ is recognized by a **top-down non-deterministic** tree automaton
(d) $L$ is generated by a **regular** tree grammar
Deterministic top-down tree automata do not recognize all regular tree languages

- Example:

$$\text{Initial}(A) = q_0$$

- $$q_0 \xrightarrow{a} (q, q)$$
- $$q \xrightarrow{b} \epsilon$$
- $$q \xrightarrow{c} \epsilon$$

Also accepts:

![Diagram of a tree automaton with rules and states](image-url)
Unranked Trees

\[ \delta(\sigma, q) \]: specified by a regular expression (i.e., regular language).

\[
\begin{array}{c}
\sigma_1 \quad \sigma_2 \quad \ldots \quad \sigma_n \\
\sigma \quad q \quad q
\end{array}
\]

\[ q_1, q_2, \ldots, q_n \in \delta(\sigma, q)? \]
Qubits:

- A quantum bit can exist in a superposition of states. In general, \( \alpha |0\rangle + \beta |1\rangle \), where \( \alpha \) and \( \beta \) are complex numbers, satisfying \( |\alpha|^2 + |\beta|^2 = 1 \).

- Any attempt to measure the state results in \( |0\rangle \) with probability \( |\alpha|^2 \), and \( |1\rangle \) with probability \( |\beta|^2 \). After the measurement, the system is in the measured state!
Quantum Entanglement

- An $n$-qubit system can exist in any superposition of the $2^n$ basis states.

$$\alpha_0|000...000\rangle + \alpha_1|000...001\rangle + \cdots + \alpha_{2^n-1}|111...111\rangle$$

- Sometimes such a state can be decomposed into the states of individual bits

$$\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) = |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

- But,

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

is not decomposable, which is called an entangled state.
Unitary Evolution

- A quantum system that is not measured (i.e. does not interact with its environment) evolves in a unitary fashion.
- That is, its evolution in a time step is given by a **unitary linear operation**.
- Such an operator is described by a matrix $U$ such that

$$UU^* = I$$

where $U^*$ is the **conjugate transpose** of $U$.

\[
\begin{pmatrix}
3 & 3 + i \\
2 - i & 2
\end{pmatrix}^* = \begin{pmatrix}
3 & 2 + i \\
3 - i & 2
\end{pmatrix}
\]
Figure: A Finite Automaton $M$

Consider the following matrix representation:

- Initial state $I = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$; final state $F = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$;
- $M_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$; $M_1 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$. 
Matrix View of DFA

The computation $q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_2$ is represented by

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

As

$$
\begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix}^T \cdot
\begin{pmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{pmatrix} \cdot
\begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix} =
\begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix}^T \cdot
\begin{pmatrix}
0 \\
1 \\
0
\end{pmatrix} = 1,
$$

the input "101" is accepted.
Quantum finite automata are obtained by letting the matrices $M_\sigma$ have complex entries. We also require each of the matrices to be unitary. E.g.

$$M_\sigma = \begin{pmatrix} -1 & 0 \\ 0 & i \end{pmatrix}$$

If all matrices only have 0 or 1 entries and the matrices are unitary, then the automaton is deterministic and reversible.
Consider the automaton in a one letter alphabet as:

\[ |\psi_0\rangle = 1 \cdot |0\rangle + 0 \cdot |1\rangle = (1, 0)^T \]

\[ M_{aa} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \]

Hence, upon reading \( aa \), \( M' \)'s state is

\[ |\psi\rangle = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = 0 \cdot |0\rangle + -1 \cdot |1\rangle \]

There are two distinct paths labelled \( aa \) from \( q_1 \) back to itself, and each has non-zero probability, the net probability of ending up in \( q_1 \) is 0.

The automaton accepts a string of odd length with probability 0.5 and a string of even length with probability 1 if its length is not a multiple of 4 and probability 0 otherwise.
The accept state of the automaton is given by an $N \times N$ projection matrix $P$, so that, given a $N$-dimensional quantum state $|\psi\rangle$, the probability of $|\psi\rangle$ being in the accept state is $\langle\psi|P|\psi\rangle = \|P|\psi\||^2$.

In the previous example, $P = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$

The probability of the state machine accepting a given finite input string $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_k)$ is given by $Pr(\sigma) = \|PU_{\sigma_k}\cdots U_{\sigma_1} U_{\sigma_0} |\psi\rangle\|^2$. In the previous example, $Pr(aa) = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}^T \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = 1$

A regular language is accepted with probability $p$ by a quantum finite automaton, if, for all sentences $\sigma$ in the language, (and a given, fixed initial state $|\psi\rangle$), one has $p < Pr(\sigma)$. 
Measure Many 1-way QFA: Measurement is performed after each input symbol is read.

Measure-many model is more powerful than the measure-once model, where the power of a model refers to the acceptance capability of the corresponding automata.

MM-1QFA can accept more languages than MO-1QFA.

Both of them accept proper subsets of regular languages.