Tree Regular Languages
Classical Formal Languages

Computational device gets a string as input.

Strings represent
\[ x \land \neg (y \lor z) \]

Vertices have labels

Labeled

Special root

Rooted, ordered, \( n \)-ary tree.

Children of a vertex ordered from left to right

\[
\begin{array}{c}
A \\
\lor \\
x \\
y \\
\lor \\
z \\
A
\end{array}
\]

Rooted tree: Every vertex has a unique path from a root.

Labeled Rooted ordered \( n \)-ary trees are

\[ t = (\text{dom}_t, \text{lab}_t) \]

\[ \text{dom}_t \subseteq \Sigma^* \]

finite strings over \( \Sigma = \{0, \ldots, n\} \)

Such that: \( \text{dom}_t \) is prefix closed

if \( u \) is prefix of \( v \) and \( v \in \text{dom}_t \) then \( u \in \text{dom}_t \)
and if $n \in \{0, \ldots, n-1\}^*$ and $j, i \in \{0, \ldots, n-1\}$
s.t. $j < i$, $u_i \in \text{dom}_t \Rightarrow u_j \in \text{dom}_t$

- $\text{lab}_t : \text{dom}_t \rightarrow \Sigma$ where $\Sigma$ is the
set of labels.

$\land \in \Sigma \quad \text{dom}_{t_0} = \Sigma \cup \{0, 1, 10, 100, 101\}$

$\\land \text{lab}_{t_0}(e) = \land \quad \text{lab}_{t_0}(0) = x$

$\\land \text{lab}_{t_0}(1) = 7 \quad \text{lab}_{t_0}(10) = \land$

$\\land \text{lab}_{t_0}(100) = y \quad \text{lab}_{t_0}(101) = \emptyset$.

**Notation** $T_n(\Sigma)$ — set of $\Sigma$-labeled, rooted, $n$-ary trees.

**Finite automata on Trees.**

**Words** $a-b-c-d-\ldots-\bigcirc$

**DFA** $q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow \ldots \rightarrow \text{root}.$

**Deterministic Tree automaton (on n-ary tree)**

$A = (Q, \Sigma, \delta, q_0, F)$

- $Q$ is a finite set of states
- $\Sigma$ is labels on the tree inputs
- $F \subseteq Q$ is a set of final states
- $S = \bigcup_{i=0}^{n} S_i$
$S_i : Q_i \times \Sigma \rightarrow A$

Run of $A$ on tree $t = (\text{dom}_t, \text{lab}_t)$ is another $A$-labeled $p = (\text{dom}_p, \text{lab}_p)$ where

- $\text{dom}_p = \text{dom}_t$.
- If $u \in \text{dom}_p$ s.t. $u$ has $i$-children
  \[
  \text{lab}_p(u) = S_i(\text{lab}_p(u_0), \ldots, \text{lab}_p(u(i-1)), \text{lab}_t(u))
  \]

**Example**

$A = (\{q_0, q_1, q_2\}, \Sigma, 0, 1, \tau, v, \land, S, \{q_0, q_1\})$

$S_0(0) = q_0$, $S_0(1) = q_1$

$S_1(q_i, \tau) = q_{i}\cdot i$

$S_2(p_1, p_2, \land) = \{q_0\}$ if $p_1 = p_2 = q_1$

$S_2(p_1, p_2, \lor) = \{q_0\}$ if $p_1 = p_L = q_0$

In all other cases, $S$ returns $\bot$. 

```
```

![Diagram](image)

```
```
On any input tree, a deterministic tree automaton has a unique run.

Acceptance: it is accepted by $A$ if the run of $A$ on $t$ is such that
the label of the root is a final state.

$L(A) = \{ t \mid A \text{ accepts } t \}$

Definition: A set of $\Sigma$-labeled $n$-ary trees is regular if there is a tree automaton $A$ such that $L(A) = K$.

Proposition: Regular tree languages are closed under complementation, intersection, and union.

Nondeterministic Tree Automata

$A = (Q, \Sigma, S, F)$

where $S_i : Q^i \times \Sigma \rightarrow 2^Q$

A run of $A$ on input $t = (\text{dom}_t, \text{lab}_t)$ is
A labeled tree \( T = (L, \text{lab}_T) \)
- \( L = \text{dom}_T \)
- \( T \) has \( i \) children
\( \text{lab}_T(u) \in \delta_i(\text{lab}_T(v_0), \ldots, \text{lab}_T(v_{i-1}), \text{lab}_T(u)) \)

**Theorem** If \( K \) is a tree language recognized by a non-deterministic tree automaton then \( K \) is regular.

**Proof** Standard subset construction.

**Proposition** Any set of finite trees is regular.

**Example** A context-free grammar
\[
G = (N, \Sigma, S, R)
\]

Nonterminals \( N \)
Terminals \( \Sigma \)
Start symbol \( S \)
Rules
\[
X \rightarrow \alpha \quad X \in N, \alpha \in (N \Sigma)^*.
\]

Parse trees of a grammar are regular.
\[
G = (N \Sigma \Sigma \ast \Sigma, N \Sigma, S, \varepsilon, \Sigma \Sigma \ast \Sigma)
\]
\[
S_0(a) = a \quad \text{if } a \in \Sigma
\]
\[
S_i(q_1 \ldots q_i, A) = A \quad \text{if } A \in N
\]
\[
\text{and } A \rightarrow q_1 \ldots q_i \in R.
\]

In all other cases, \( S \) returns \( \ast \).