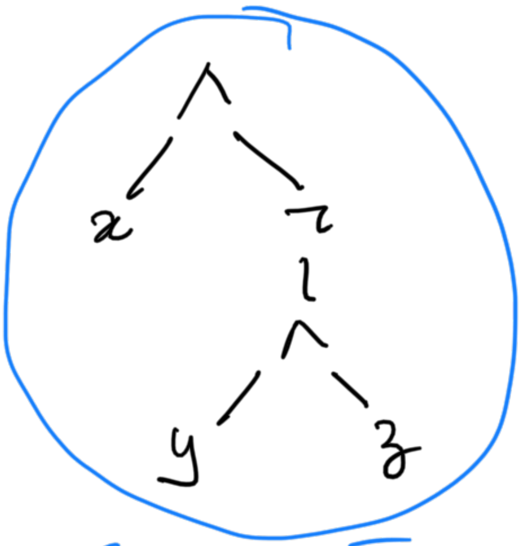
Tree Regular Languages

Classical Formal Languages: Computational device gets a string as input.

Strings represent

$x \wedge \neg(y \wedge z)$ —



Vertices have labels

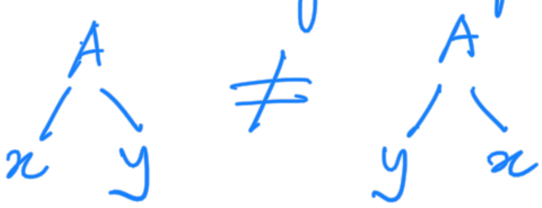
Labeled

Special root

Every vertex $\leq n$ children

Rooted, **ordered**, **n-ary** trees.

Children of a vertex ordered from left to right



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

Labeled Rooted ordered n-ary trees are

$t = (dom_t, lab_t)$

- $dom_t \subseteq \{0, \ldots n-1\}^*$

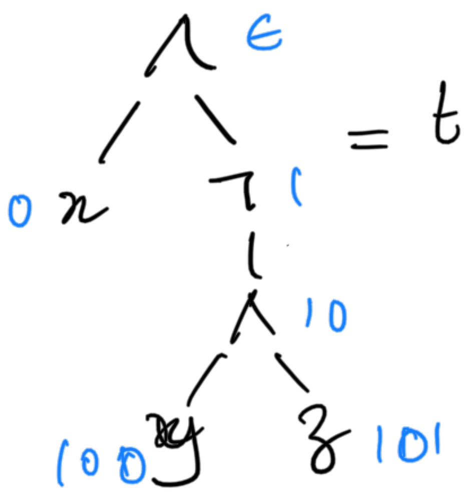
finite strings over $\{0, \ldots n-1\}$

such that dom_t is **prefix closed**

if u is prefix of v and $v \in dom_t$ then $u \in dom_t$.

and $\forall u \in \{0,\dots,n-1\}^*$ and $j, i \in \{0,\dots,n-1\}$ s.t $j<i$, $ui \in dom_t \Rightarrow uj \in dom_t$

- $lab_t : dom_t \to \Sigma$ where Σ is the set of labels.



$= t_0$

$dom_{t_0} = \{\epsilon, 0, 1, 10, 100, 101\}$

$lab_{t_0}(\epsilon) = \wedge \quad lab_{t_0}(0) = x$

$lab_{t_0}(1) = \neg \quad lab_{t_0}(10) = \wedge$

$lab_{t_0}(100) = y \quad lab_{t_0}(101) = z.$

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

Finite automata on Trees.

Words $\quad a - b - c - d - \dots - \textcircled{z}$ root.

DFA $\quad q_1 \to q_2 \to q_3 \;\dots$

Deterministic Tree automaton (on n-ary trees)

$\mathcal{A} = (Q, \Sigma, \delta, F)$

- Q is a finite set of states
- Σ is labels on the tree inputs
- $F \subseteq Q$ is a set of final states
- $\delta = \bigcup_{i=0}^{n} \delta_i$

$$\delta_i : Q^i \times \Sigma \to Q$$

Run of $\mathcal{A}$ on tree $t = (dom_t, lab_t)$ is another Q-labeled $\rho = (dom_\rho, lab_\rho)$ where

- $dom_\rho = dom_t$.
- $\forall u \in dom_\rho$ s.t u has i-children

$$lab_\rho(u) = \delta_i\big(lab_\rho(u0), \dots, lab_\rho(u(i-1)), lab_t(u)\big)$$

Example $\mathcal{A} = (\{q_0, q_1, r\}, \{0, 1, \neg, \vee, \wedge\}, \delta, \{q_1\})$

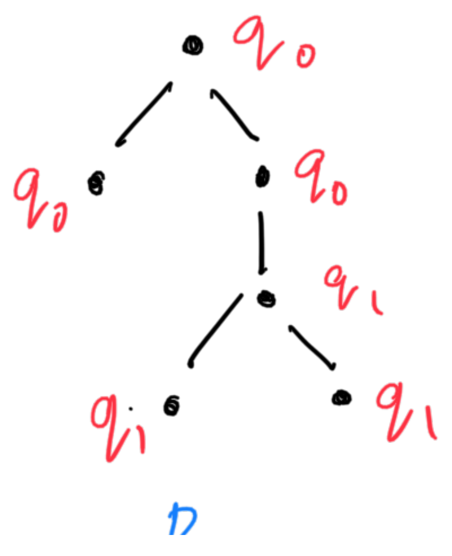
$\delta_0(0) = q_0 \quad \delta_0(1) = q_1$

$\delta_1(q_i, \neg) = q_{1-i}$

$$\delta_2(p_1, p_2, \wedge) = \begin{cases} q_1 & \text{if } p_1 = p_2 = q_1 \\ q_0 & \text{o.w.} \end{cases}$$

$$\delta_2(p_1, p_2, \vee) = \begin{cases} q_0 & \text{if } p_1 = p_2 = q_0 \\ q_1 & \text{o.w.} \end{cases}$$

In all other cases, δ returns r.



Input   Run

$\wedge$ r
$|$ $|$
0 q_0

Input   Run

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

Acceptance t is accepted by $\mathcal{A}$ if the run of A on t is such that the label of the root is a final state.

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

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

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

Nondeterministic Tree Automaton

$\mathcal{A} = (Q, \Sigma, \delta, F)$

where $\delta_i : Q^i \times \Sigma \to 2^Q$

A Run of $\mathcal{A}$ on input $t = (dom_t, lab_t)$ is

A run of ...

Q - labeled tree $\rho = (dom_\rho, lab_\rho)$

- $dom_\rho = dom_t$
- $\forall u$ with i-children

$lab_\rho(u) \in \delta_i(lab_\rho(u0) \dots lab_\rho(u(i-1)), lab_t(u))$

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

$$G = (\underbrace{N}_{\text{nonterminals}}, \underbrace{T}_{\text{terminals}}, \underbrace{S}_{\text{start sym}}, \underbrace{R}_{\text{Rules}})$$

$X \to \alpha$

$X \in N, \alpha \in (N \cup T)^*$.

Parse trees of a grammar is regular.

$\mathcal{A} = (N \cup T \cup \{*\}, N \cup T, \delta, \{S\})$

$\delta_0(a) = a$ if $a \in T$.

$\delta_i(q_1 \dots q_i, A) = A$ if $A \in N$ and $A \to q_1 \dots q_i \in R$.

In all other cases, δ returns $*$.