Recapitulation: Hedge Automata for XML Languages

$\Sigma = \{a, b, c\}$

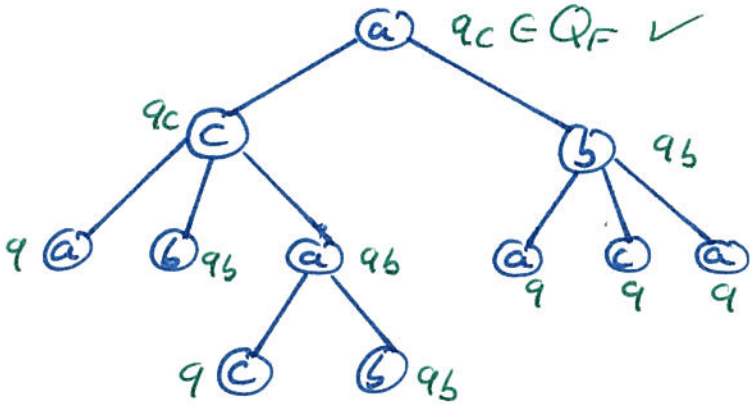
$H = (Q, \to, Q_F)$, $Q = \{q, q_b, q_c\}$, $Q_i = \{q_c\}$

Transitions:

$Q^* \to_{a|c} q$ $\quad Q^* q_b Q^* \to_{a|c} q_b$ $\quad Q^* q_c Q^* \to_{a|b|c} q_c$

$Q^* \to_b q_b$ $\quad Q^* q_b Q^* q_b Q^* \to_c q_c$

Run on



9.2 Membership

Goal: Given a DTD D and an XML document t.

Decide $t \in L(D)$ $(= L(H_D))$.

Theorem:

Membership for NHTs with horizontal languages represented by NFAs can be decided in polynomial time (in the size of the tree and the size of the automaton).

Algorithm:

Given · NHT $H = (Q, \to, Q_F)$

· tree $t: T \to \Sigma$.

Construct a mapping

$S: T \to \mathbb{P}(Q)$. (S like run)

↳ It labels each node $w \in T$
with the set of states $\rho(w) \subseteq Q$
that are reachable at w
in a run of $\mathcal{A}$ on t.

↳ To this end, pick a node $w \in T$

- whose children $w.0, \dots, w.(n-1)$ are already labelled by $\rho(w.0) = Q_0, \dots, \rho(w.(n-1)) = Q_{n-1}$.
- Check for each transition $R \to a\, q$ with $a = t(w)$ whether it can be applied to a string $q_0 \dots q_{n-1}$ obtained from selecting $q_i \in Q_i$.

The algorithm:

input: NFA $\mathcal{A} = (Q, \to, Q_F)$, tree $t: T \to \Sigma$

begin:

set $\rho(w) := \bot$ f.a. $w \in T$.

while ($\exists\, w \in T$ with $\rho(w) = \bot$ and
$\rho(w.i) \neq \bot$ f.a. children $w.0, \dots, w.(n-1)$ of w) {

$M := \emptyset$

for each transition $R \to a\, q$ with $a = t(w)$ {

if ($\exists\, q_0 \in \rho(w.0), \dots, q_{n-1} \in \rho(w.(n-1))$
with $q_0 \dots q_{n-1} \in R$) {

$M := M \cup \{q\}$

}

}

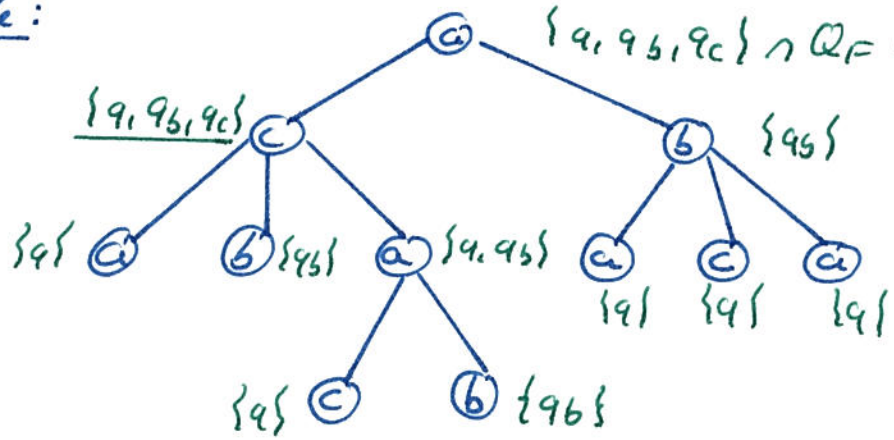
$\rho(w) := M$

}

output "yes" if $\rho(\varepsilon) \cap Q_F \neq \emptyset$, "no" otherwise.

end

Example:



Why is ⓒ labelled by $\{q, q_b, q_c\}$?

$Q^* \xrightarrow{c} q$ applies $\rightsquigarrow M = \emptyset \cup \{q\}$

$Q^* q_b Q^* \xrightarrow{c} q_b$ applies $\rightsquigarrow M = \{q\} \cup \{q_b\}$

$Q^* q_b Q^* q_b Q^* \xrightarrow{c} q_c$ applies $\rightsquigarrow M = \{q, q_b\} \cup \{q_c\}$ ✓

The only nonpolynomial check:

↳ May be if $(\exists q_0 \in \delta(\omega, 0), \ldots$ inside the loop.

↳ Works in polynomial time as follows:

- Start in initial state of the NFA for the horizontal language R.
- Collect all NFA states reachable (in a single step) with NFA states in $\delta(\omega, 0) = Q_0$. (transition labels)
- Compute from these all NFA states reachable with elements in $\delta(\omega, 1) = Q_1$.
- etc.
- If last set contains a final state of the NFA, condition is satisfied.

↳ Algorithm can be understood as "powerset construction along $Q_0, \ldots, Q_{n-1}$"

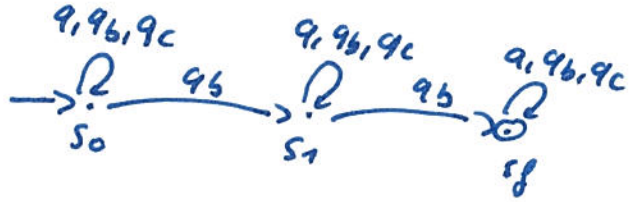
Example (continued):

Q_0 = $\{q_1\}$, Q_1 = $\{q_5\}$, Q_2 = $\{q_1, q_5\}$.

- Consider node ⓒ with child sequence $\{q_1\}, \{q_5\}, \{q_1, q_5\}$.
- Consider transition $Q^* q_5 Q^* q_5 Q^* \to q_c$.

↳ The regular expression is represented by

$\mathcal{A}_H$



↳ To check whether

$\exists\, q_0 \in \{q_1\}, q_1 \in \{q_5\}, q_2 \in \{q_1, q_5\}$

with $q_0 q_1 q_2 \in L(\mathcal{A}_H) = Q^* . q_5 . Q^* . q_5 . Q^*$

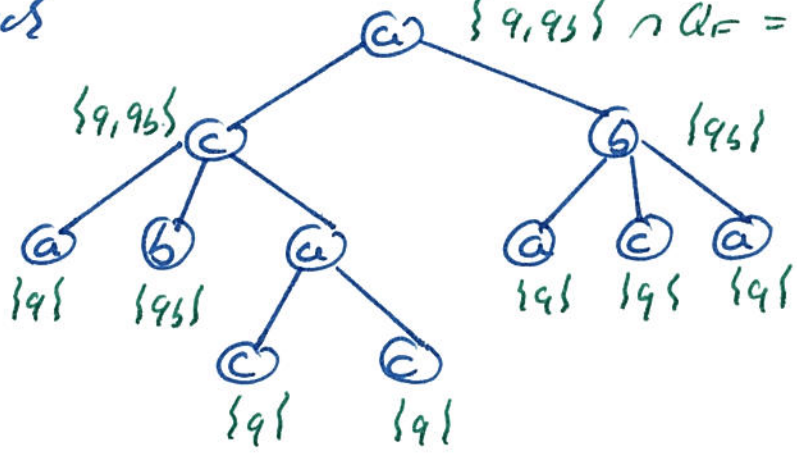
apply the above procedure:

$$\{s_0\} \xrightarrow{\{q_1\}} \{s_0\} \xrightarrow{\{q_5\}} \{s_0, s_1\} \xrightarrow{\{q_1, q_5\}} \{s_0, s_1, s_f\}$$

Since $s_f \in \{s_0, s_1, s_f\}$, add q_c to M.

Example:

Type check



More on XML documents:

↳ Work on data inside documents

↳ Adapt schema representations of different companies

→ Deal with inclusion, renaming, structure changes

→ Algorithmic support is needed.