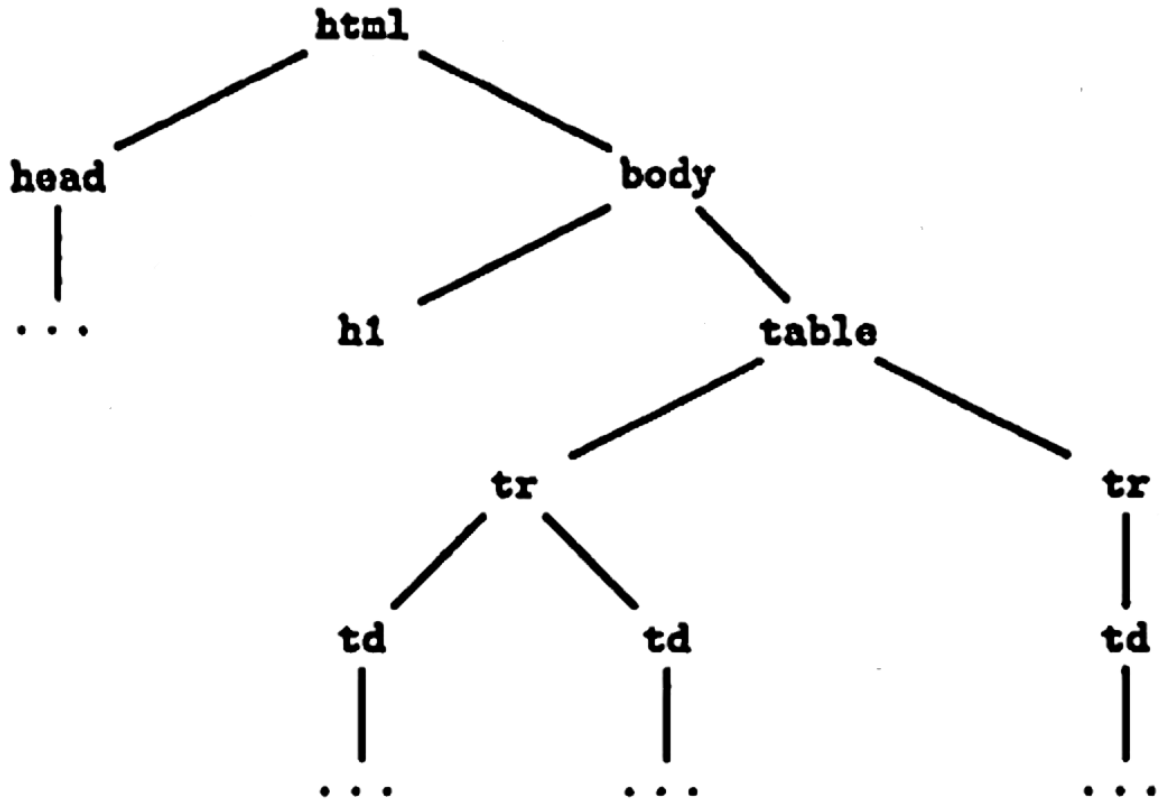
Automata for unranked trees

- XML: regarded as unranked trees

```
<html>
    <head>
        ...
    </head>
    <body>
        <h1>...</h1>
        <table>
            <tr>
                <td>...</td>
                <td>...</td>
            </tr>
            <tr>
                <td>...</td>
            </tr>
        </table>
    </body>
</html>
```

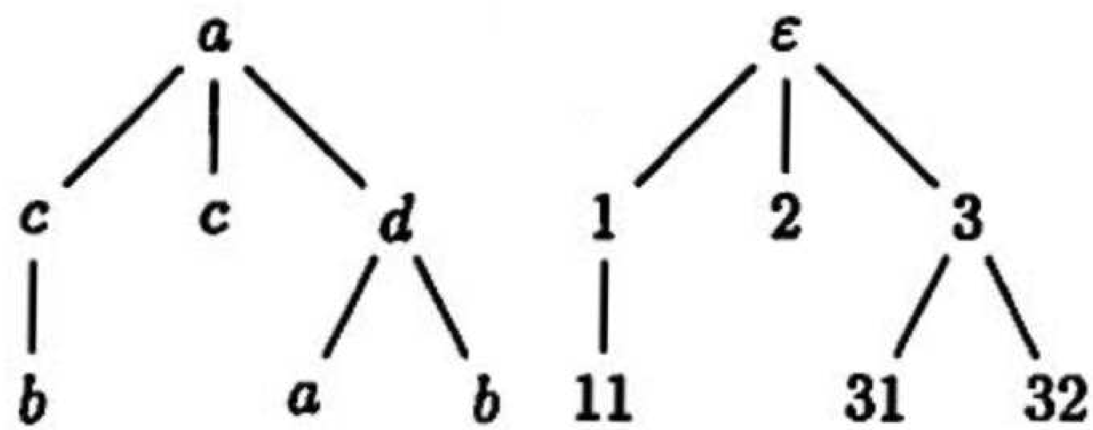


- **Possible choices for handling unranked trees**
  - **Coding them as ranked trees**

    Complex due to the structure
  - **Handling without coding**

    New technique is necessary to handle unranked trees

- Set Σ of unranked function symbols
- **unranked tree**: $\mathsf{H}(\Sigma)$
  - $t_1 \cdots t_n$ is called a **hedge** for $n \geq 0$, $t_1, \ldots, t_n \in \mathsf{H}(\Sigma)$
  - $a(h) \in \mathsf{H}(\Sigma)$ for a hedge h and $a \in \Sigma$
- Ex.: $t = a(c(b)cd(ab))$



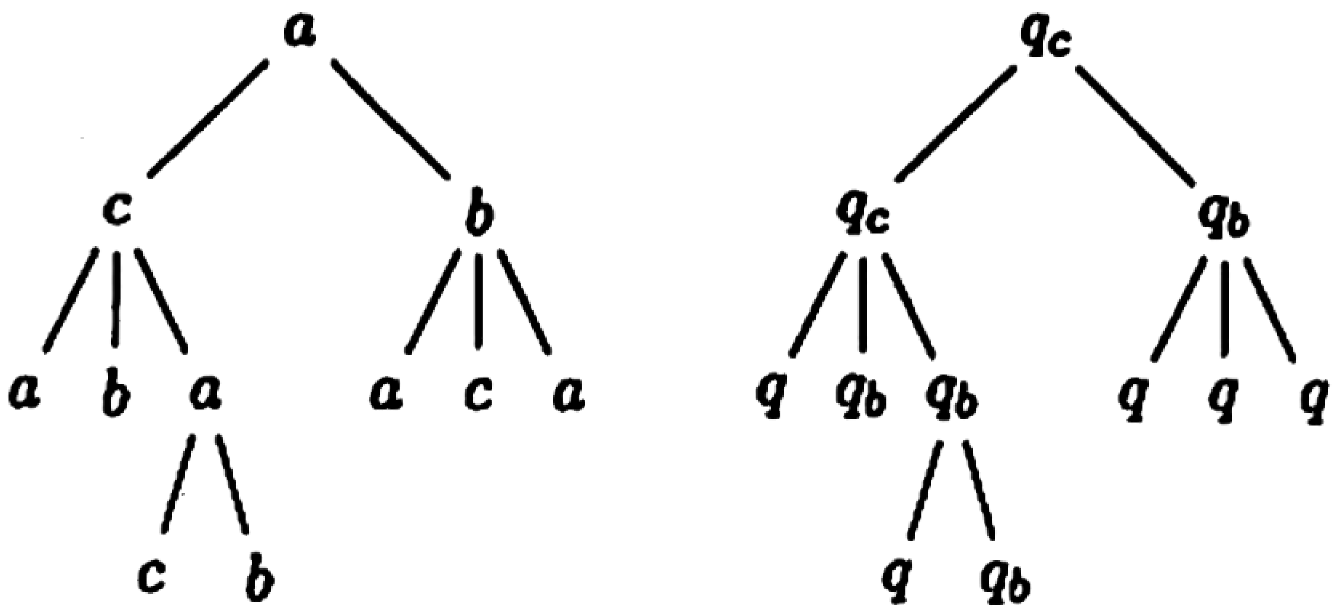
unranked tree   position tree

- **Non-deterministic finite hedge automaton(NFHA)**
  Similar to NFTA except for transition rules
  $A = (Q, \Sigma, Q_f, \Delta)$
- **Ex. of transition rules**
  - $a((q_a q_b)^*) \to q$,
    which represents the following rules
    $a \to q$, $a(q_a q_b) \to q$, $a(q_a q_b q_a q_b) \to q$, $\ldots$

- **Ex. of complete NFHA: accepting trees containing two b's whose nearest common ancestor is c**

  - $A = (Q, \{a, b, c\}, \{q_c\}, \Delta)$, $\quad Q = \{q, q_b, q_c\}$

$$
\begin{array}{lll}
a(Q^*) \to q & a(Q^* q_b Q^*) \to q_b & a(Q^* q_c Q^*) \to q_c \\
b(Q^*) \to q_b & c(Q^* q_b Q^*) \to q_b & b(Q^* q_c Q^*) \to q_c \\
c(Q^*) \to q & c(Q^* q_b Q^* q_b Q^*) \to q_c & c(Q^* q_c Q^*) \to q_c
\end{array}
$$



- **Ex. of incomplete NFHA: accepting true Boolean-expression**
  - $A = (Q, \{0, 1, not, and, or\}, \{q_1\}, \Delta)$

    $Q = \{q_0, q_1\}$

    $$\Delta = \begin{cases} 0(\varepsilon) \to q_0, & or(Q^* q_1 Q^*) \to q_1, \\ 1(\varepsilon) \to q_1, & or(q_0 q_0^*) \to q_0, \\ not(q_0) \to q_1, & and(Q^* q_0 Q^*) \to q_0, \\ not(q_1) \to q_0 & and(q_1 q_1^*) \to q_1 \end{cases}$$
- **Completion of NFHA is possible**

- **Deterministic finite hedge automaton (DFHA)**: NFHA satisfying that
  - For any rules $a(R_1) \to q_1$ and $a(R_2) \to q_2$, $R_1 \cap R_2 \neq \emptyset$ implies $q_1 = q_2$
- Determination of NFHA is possible (there exists an equivalent DFHA)
- NFHA inherits properties of ranked automaton, because encoding them as ranked trees is possible
  - Closed union, intersection and complement
  - Emptiness problem is decidable