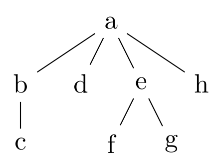
# Foundations of XML Types

## Trees and Tree Automata

1. Every tree can be represented as a binary tree. Give the binary tree associated with the tree shown below by the encoding of trees into binary trees used during the course:




2. Give a bottom-up deterministic tree automaton that recognize the tree language $L$ composed of the two trees below:


3. Bottom-up tree automata seen during the course traverse trees from the leaves to the root. In a similar manner, one may define top-down tree automata that recognize trees by going in the opposite direction: from the root to the leaves. Specifically, a top-down tree automaton $A$ consists in:

| | |
|---|---|
| Alphabet($A$): | finite alphabet of symbols |
| States($A$): | finite set of states |
| Rules($A$): | finite set of transition rules |
| Initial($A$): | finite set of initial states ($\subseteq$ States($A$)) |
| $q_{\text{acc}} \in$ States($A$) : | final state |

There are two major differences with automata seen during the course:

- transition rules are either of the form: $q \xrightarrow{a} (q_1, q_2)$ where $q, q_1, q_2 \in$ States($A$) and $a \in$ Alphabet($A$) or of the form $q \xrightarrow{\epsilon} q_1$ for leaves.
- a tree is accepted if and only if there exists a run for which all the leaves are labeled with $q_{\text{acc}}$.

Give a top-down tree automaton that recognizes $L$.

4. Do you see any interest of top-down tree automata in the context of XML stream processing where XML documents are sequentially parsed (only once) and processed on the fly? Explain.

5. A top-down tree automaton is deterministic iff (1) there is at most one initial state and (2) for each $q \in$ States($A$) et $a \in$ Alphabet($A$) there is at most one rule $q \xrightarrow{a} (q_1, q_2)$ (intuitively, there is at most one possible transition for each state and symbol).
Is it possible to give a deterministic top-down tree automaton that recognize $L$? Either give it or justify.

6. It is known that non-deterministic bottom-up and non-deterministic top-down automata are equally expressive. From your answers to the previous questions, what can you conclude about the respective expressive power of deterministic bottom-up and deterministic top-down tree automata? Justify.