Lesson 6: Tree automata

Theme: Tree automata as a model of computation for processing trees.

1 Labelled binary tree

Let Σ be a finite set of symbols. A Σ-binary-tree is a (rooted) tree $T = (r, I, F, \ell ab)$ where:

- r is the root, I is the set of vertices, and F is the set of edges;
- each vertex has either two children, called the left and child, respectively, or no children;
- each vertex $i \in I$ is labelled with a symbol $\ell ab(i)$ from Σ.

Those vertices with no children are called *leaf* vertices. When it is clear from the context, we usually simply write tree, instead of Σ-binary-tree.

2 Top-down tree automata

A *top-down non-deterministic tree automaton* over Σ is a tuple $\mathcal{M} = (Q, Q_0, \Delta)$, where:

- Q is a finite set of states;
- $Q_0 \subseteq Q$ is the set of *initial* states;
- $\Delta \subseteq (Q \times \Sigma) \cup (Q \times \Sigma \times Q \times Q)$ is a finite set of transitions.

We usually write each transition $(p, a) \in \Delta$ as $(p, a) \to \top$ and $(p, a, q_1, q_2) \in \Delta$ as $(p, a) \to (q_1, q_2)$. The automaton $\mathcal{M}$ is *deterministic* for every $(p, a) \in Q \times \Sigma$, there is exactly one $(q_1, q_2) \in Q \times Q$ such that $(p, a) \to (q_1, q_2) \in \Delta$, and for every $a \in \Sigma$, there is exactly one $(p, a) \in Q \times \Sigma$.

A tree $T = (r, I, F, \ell ab)$ is *accepted* by $\mathcal{M} = (Q, Q_0, \Delta)$, if we can relabelled the vertices in $T = (r, I, F, \ell ab)$ with the states in Q such that:

- the root r is relabelled with one of the initial state in Q_0;
- the relabelling of other vertices is done top-down according to the transitions in Δ, i.e.,
  - for every interior vertex $i \in I$, if it is relabelled with state p, then its children are labelled with q_1, q_2 such that there is a transition $(p, \ell ab(i)) \to (q_1, q_2)$ in Δ;
  - every leaf vertex $i \in I$ can be relabelled with state p, if there is a transition $(p, \ell ab(i)) \to \top$ in Δ;

We denote by $L(\mathcal{M})$ the set of all trees accepted by $\mathcal{M}$.

3 Bottom-up tree automata

A *bottom-up non-deterministic tree automaton* over Σ is a tuple $\mathcal{M} = (Q, Q_f, \Delta)$, where:

- Q is a finite set of states;
- $Q_f \subseteq Q$ is the set of *final* states;

- Δ is a finite set of transitions of the form:

$$\begin{aligned} a &\rightarrow q \\ (p_1, a_1, p_2, a_2) &\rightarrow q \end{aligned}$$

  where $a, a_1, a_2 \in \Sigma$ and $p_1, p_2, q \in Q$. Transitions of the former form are called the initial rules.

The automaton $\mathcal{M}$ is *deterministic* for every label $a \in \Sigma$, there is exactly one $q \in Q$ such that $a \rightarrow q \in \Delta$; and for every (p_1, a_1, p_2, a_2), there is exactly one $q \in Q$ such that $(p_1, a_1, p_2, a_2) \rightarrow q \in \Delta$.

A tree $T = (r, I, F, \ell ab)$ is *accepted* by $\mathcal{M} = (Q, Q_0, \Delta)$, if we can relabelled the vertices in $T = (r, I, F, \ell ab)$ with the states in Q such that:

- the root r is relabelled with one of the final states in Q_f;
- the relabelling is done bottom-up according to the transitions in Δ, i.e.,
  - every leaf vertex can be relabelled with $q \in Q$, if there is a transition $\ell ab(i) \rightarrow q \in \Delta$;
  - for every interior vertices $i_1, i_2 \in I$ with the same parent, if they are relabelled with state p_1, p_2, respectively, then their parent can be relabelled with q, if there is a transition $p_1, \ell ab(i_1), p_2, \ell ab(i_1) \rightarrow q$ in Δ.

We denote by $L(\mathcal{M})$ the set of all trees accepted by $\mathcal{M}$.

4 Non-deterministic vs. deterministic tree automata

Theorem 6.1

- *Non-deterministic bottom-up tree automata = deterministic bottom-up tree automata = non-deterministic top-down tree automata.*
- *However, deterministic top-down tree automata are weaker than their non-deterministic counterpart.*

Appendix

For more details about tree automata, please consult [1].

References

[1] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2007. release October, 12th 2007.