Recapitulation:

Bottom-up tree automata:

$A = (Q, \to, Q_F)$

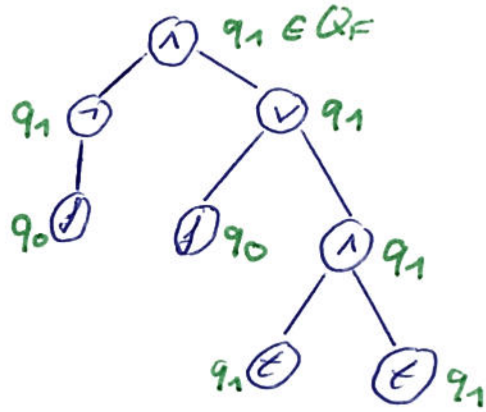
Run of A on tree $t: T \to \Sigma$ is

$r: T \to Q$

so that

$$(\underbrace{r(w.0)}_{q_0}, \dots, \underbrace{r(w.(n-1))}_{q_{n-1}}) \to_a \underbrace{r(w)}_{q}$$

where $a = t(w)$ and $n = rk(a)$.



Top-down tree automata:

$A = (Q, q_i, \to)$

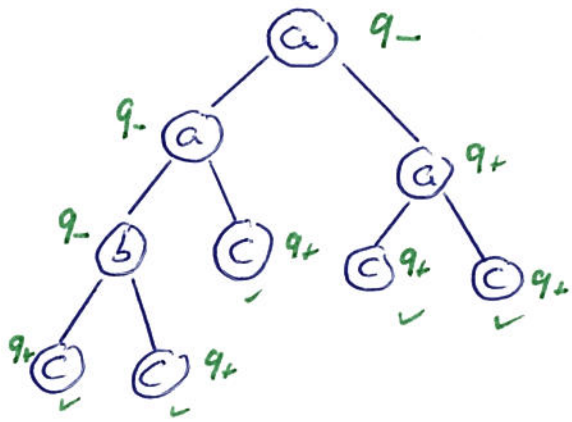
Run of A on tree $t: T \to \Sigma$ is

$r: T \to Q$

so that

$$r(w) \to_a (r(w.0), \dots, r(w.(n-1)))$$

with $a = t(w)$ and $n = rk(a)$.



- Are TDTA and BUTA equally expressive? Yes.
- Are deterministic TDTA as powerful as nondeterministic TDTA? No

Theorem (TDTA accept precisely the regular tree languages):
A tree language $L \subseteq T_\Sigma$ is accepted by a BUTA iff it is accepted by a TDTA.

Idea: Nondeterminism

If $(q_0, \dots, q_{n-1}) \to_a q$ in BUTA then TDTA guesses

$$q \to_a (q_0, \dots, q_{n-1}).$$

And vice versa.

Behind this

$Q^n \times Q = Q \times Q^n$.

Construction:

- Let $A = (Q, q_i, \rightarrow)$ a TDTA.

  Then $A' = (Q, \rightarrow', \{q_i\})$ with

  $$(q_0, \dots, q_{n-1}) \rightarrow'_a q \quad \text{if} \quad q \rightarrow_a (q_0, \dots, q_{n-1})$$

  is a BUTA with $L(A) = L(A')$.

- For the reverse direction, consider a BUTA

  $$A = (Q, \rightarrow, Q_F).$$

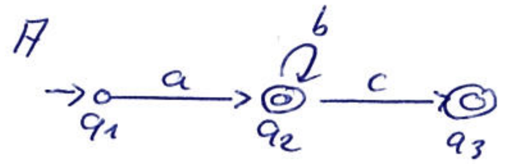
  Note that wlog. we can assume the final states to contain only one element:

  $$Q_F = \{q_{fin}\}.$$

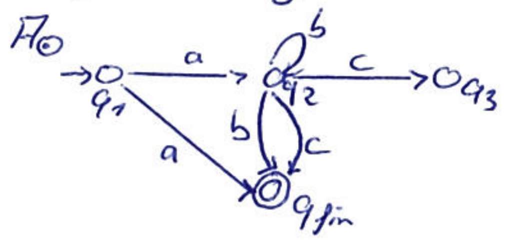
  Why?

  Guess (last) transition into final state.

  Illustration on finite automata:



  Replaced by



  with $L(A) = L(A_\odot)$.

  In general:

  - Add new final state
  - Turn previous final states non-final
  - For every edge $q \xrightarrow{a} q'$ to a previous final state, add a copy $q \xrightarrow{a} q_{fin}$ to new final state.

  (guess last transition).

  Construction can be extended to BUTA.

For such a BUTA $A = (Q, \rightarrow, \{q_{fin}\})$
the corresponding TDTA is $A' = (Q, q_{fin}, \rightarrow')$
with

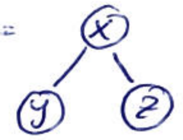
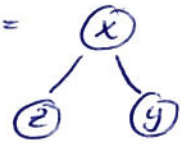
$$q \rightarrow'_a (q_0, \ldots, q_{n-1}) \quad \text{if} \quad (q_0, \ldots, q_{n-1}) \rightarrow_a q.$$

For both constructions, $L(A) = L(A')$.

□

<u>Theorem</u>:
There are regular tree languages that cannot be accepted by a DTDTA.

<u>Proof</u>:
Consider $\Sigma = \{x/2, y/0, z/0\}$.
Let $L = \{t_0, t_1\}$ with

$t_0 =$  and $t_1 =$ .

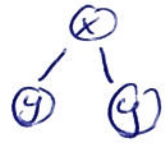
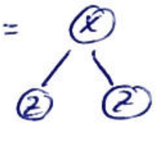
↳ This language can be accepted by a TDTA.
↳ Towards a contradiction, assume $A = (Q, q_i, \rightarrow)$ is a DTDTA that accepts L, i.e., $L = L(A)$.
Since $t_0 \in L$, there are states $q_1, q_2 \in Q$ with

$$q_i \rightarrow_x (q_1, q_2).$$

Moreover, $q_1 \rightarrow_y$ and $q_2 \rightarrow_z$.
But as also $t_1 \in L(A)$, we have

$$q_1 \rightarrow_z \quad \text{and} \quad q_2 \rightarrow_y.$$

(note that we cannot choose a different transition $q_i \rightarrow_x \ldots$ because the automaton is deterministic).
As a consequence, we additionally get

$t_2 =$  and $t_3 =$  $\in L(A)$. A contradiction to $L(A) = L$.

□

8.1 Decision procedures

↳ Tree automata as a tool to obtain decidability results for

- logics
- other operational models.

↳ Key problem: Emptiness.

Theorem:

The emptiness problem $L(A) = \emptyset$ for a BUTA $A = (Q, \rightarrow, Q_F)$ can be solved in time $O(|\rightarrow|)$.

Construction:

Build monotonically increasing sequence

$$R_0 \subseteq R_1 \subseteq \dots$$

of sets of states $R_i \subseteq Q$ that are reachable in up to i-steps (applications of transitions).

More precisely,

$$R_0 := \emptyset$$

$$R_{i+1} := R_i \cup \bigcup_{a \in \Sigma} \{ q \in Q \mid \text{there are } q_0, \dots, q_{n-1} \in R_i \text{ so that } (q_0, \dots, q_{n-1}) \rightarrow_a q \text{ with } n = rk(a) \}$$

Sequence reaches a fixed point after at most $|Q|$-steps.

Let $R = R_i$ with $R_i = R_{i+1}$.

Claim:

$L(A) \neq \emptyset$ iff $R \cap Q_F \neq \emptyset$.

Example:

Consider following BUTA that finds trees with at least one b.

$\Sigma = \{ a/2, b/2, c/0 \}$

$A = (\{q_y, q_n\}, \rightarrow, \{q_y\})$

and

$\rightarrow_c q_n \qquad (q_n, q_n) \rightarrow_a q_n \qquad (*, *) \rightarrow_b q_y$

$(q_n, q_y) \rightarrow_a q_y$

$(q_y, q_n) \rightarrow_a q_y$

$(q_y, q_y) \rightarrow_a q_y$

Here we have

$R_0 = \emptyset \quad R_1 = \{q_n\} \quad R_2 = \{q_n, q_y\} = R_3$

<u>Proof (of the emptiness theorem):</u>

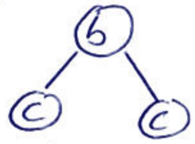
"$\Leftarrow$" <u>Idea:</u>

- Let $q \in R \cap Q_F$.
- Then $q \in R = R_k$ because there is a run of A on some tree of height $\leq k$ that labels the root by q.
  (select the tree according to the transitions that add the states to the R_i).
- Reconstructing the tree yields $L(A) \neq \emptyset$ (because $q \in Q_F$).
  (Indeed, one can show that if q is added in R_k, then the tree has height k.)

<u>In the example:</u>

- $q_n \in R_1$ by $\to_c q_n$

  Reconstruct c
- $q_y \in R_2$ by $(q_n, q_n) \to_b q_y$

  Reconstruct 

  where c is the tree for q_n.

<u>By induction:</u>

Prove for every $k \in \mathbb{N}$ the following.

If $q \in R_k$ then there is a tree t_q of height $\leq k$ that
- admits a run of A so that
- the root of t_q is labelled by q.

<u>IA:</u> $R_1 = \bigcup_{a \in \Sigma} \{q \in Q \mid \to_a q\}$

For q with $\to_a q$ select $t_q = a$.

<u>IS:</u> Assume the claim holds for the states in R_k and consider $q \in R_{k+1} \setminus R_k$.

(if $q \in R_h$ then the claim holds by the hypothesis).

Since q is added in R_{h+1}, there are states

$$q_0, \dots, q_{n-1} \in R_h \quad \text{with} \quad (q_0, \dots, q_{n-1}) \to_a q$$

for some $a \in \Sigma$ with $n = rk(a)$.

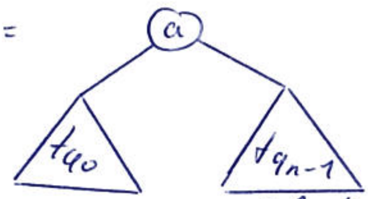
By the hypothesis, we find trees

$$t_{q_0}, \dots, t_{q_{n-1}}$$

of height $\le h$ that

- have runs $r_0, \dots, r_{n-1}$ which
- label the root of t_{q_i} by q_i.

For q we select

$t_q =$ 

of height $\le h+1$.

It admits a run labelling the root by q:

- reuse $r_0, \dots, r_{n-1}$ on $t_{q_0}, \dots, t_{q_{n-1}}$
- apply $(q_0, \dots, q_{n-1}) \to_a q$ as a last step.

"$\Rightarrow$" Let $t : T \to \Sigma$ in $L(A)$.

Let r an accepting run of A on t.

↳ If q occurs at a subtree of height h as root then $q \in R_h$.

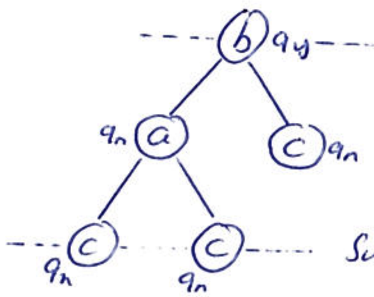
↳ Apply this to the state q_{fin} labelling the root of t:

- $q_{fin} \in Q_F$ by acceptance and
- $q_{fin} \in R_{\text{height of } t}$

↳ As $R_{\text{height of } t} \subseteq R$ we have

$$R \cap Q_F \neq \emptyset.$$

In the example:



Subtree of height 3, $q_y \in R_3$

Subtrees of height 1, $q_n \in R_1$

The above reasoning only yields $|Q| | \rightarrow |$ as time.
Observe that it is sufficient to use every transition only once. $\square$

Corollary:

Emptiness for TDTA with n transitions can be solved in time $O(n)$.

For universality, use reduction to emptiness via complementation.

Theorem:

For a regular tree language L over Σ (represented by BUTA or TDTA) the universality problem $L = T_\Sigma$ is decidable.

Proof:

Assume $L = L(A)$ for some BUTA.
Construct DBUTA A' with $L(A) = L(A')$.
Complement language by swapping final states:

$$L(\overline{A'}) = \overline{L(A')}.$$

Consequence:

$$L(\overline{A'}) = \overline{L} \text{ and thus } L = T_\Sigma \text{ iff } \overline{L} = \emptyset$$
$$\text{iff } L(\overline{A'}) = \emptyset$$

Emptiness has just been shown to be decidable. $\square$

Another problem: inclusion

$L(A) \subseteq L(B)$ for regular tree languages represented by BUTA or TDTA.

We have

$L(A) \subseteq L(B)$ iff $L(A) \cap \overline{L(B)} = \emptyset$.

Problem is decidable if regular tree languages are closed under intersection.

Lemma:

Regular tree languages are closed under intersection.

Proof:

- $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$
- Regular tree languages are closed under $\overline{}$ and $\cup$.
- Direct construction is possible and similar to word automata.

$\square$

Corollary:

Inclusion $L(A) \subseteq L(B)$ and equivalence $L(A) = L(B)$ are decidable for regular tree languages.

Consequences of these results:

↳ Define a WMSO on finite trees (with several succ predicates $succ_0, \ldots, succ_{maxrk}$)

↳ Will have a decidable satisfiability problem

↳ Supported by tool Mona, Aarhus, Denmark.