Automata theory and its applications

Lecture 17 -18: Automata over unranked trees

Zhilin Wu

State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences

April 3, 2013

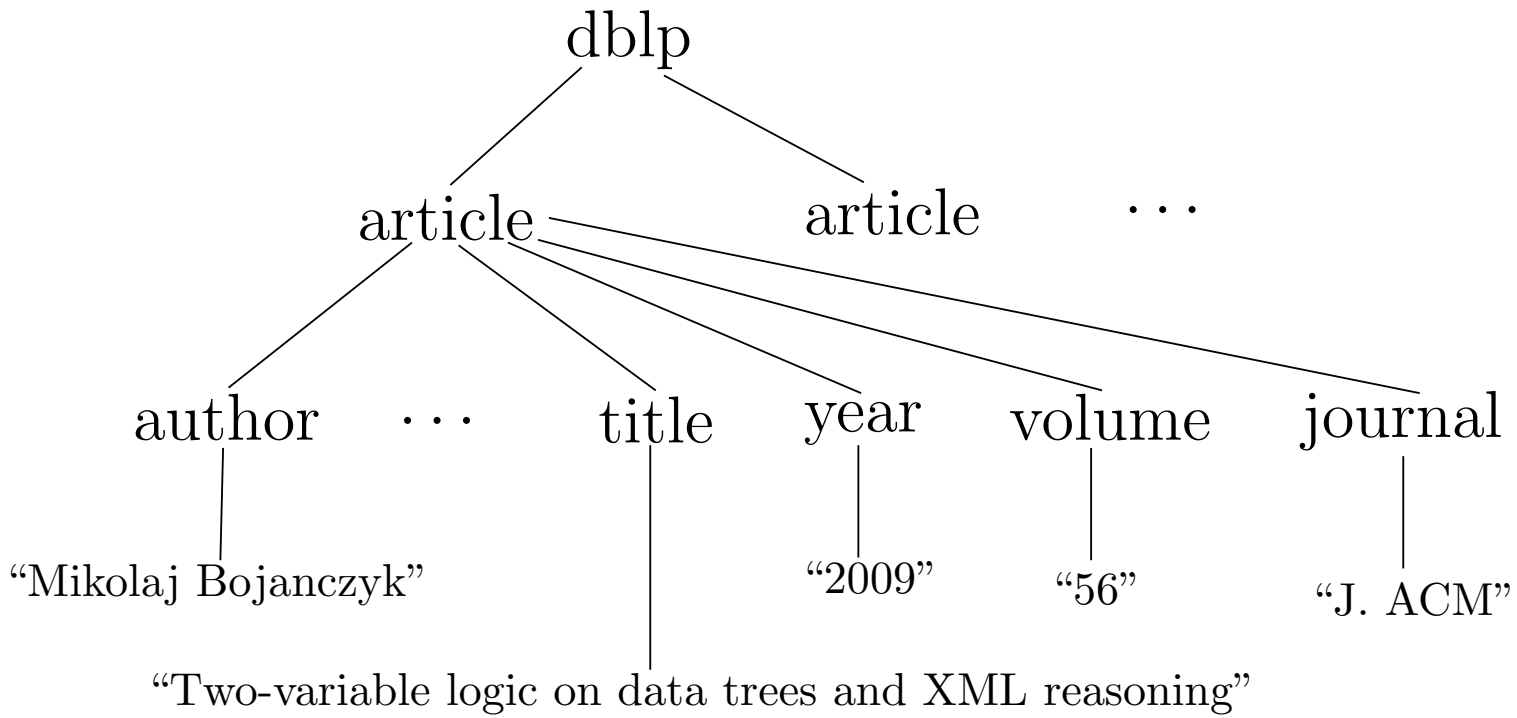
Outline

1. Hedge automata
2. Closure properties
3. Minimization
4. Decision problems

Unranked trees: The motivation

```
<dblp>
  <article key="journals/jacm/BojanczykMSS09" mdate="2009-05-20">
    <author>Mikolaj Bojanczyk</author>
    <author>Anca Muscholl</author>
    <author>Thomas Schwentick</author>
    <author>Luc Segoufin</author>
    <title>
      Two-variable logic on data trees and XML reasoning.
    </title>
    <year>2009</year>
    <volume>56</volume>
    <journal>J. ACM</journal>
    <number>3</number>
    <ee>http://doi.acm.org/10.1145/1516512.1516515</ee>
    <url>db/journals/jacm/jacm56.html#BojanczykMSS09</url>
  </article>
</dblp>
```

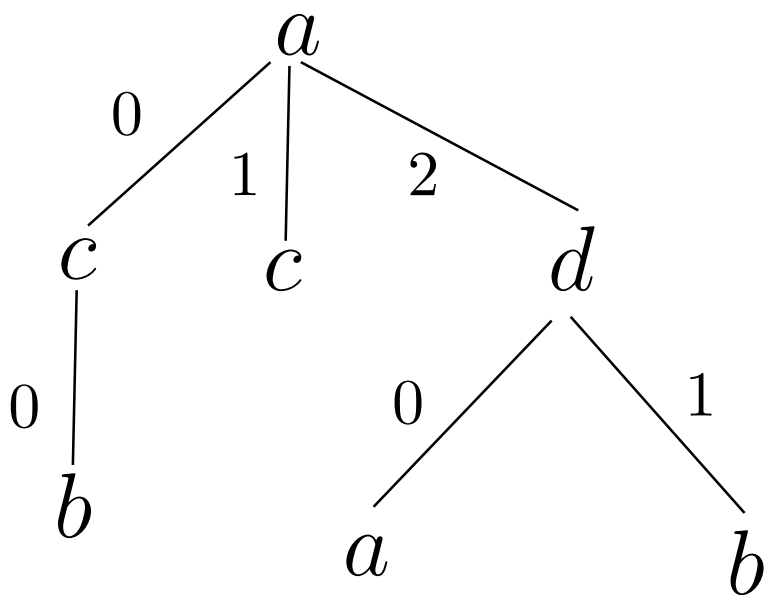
Unranked trees: The motivation



Unranked trees: The definition

An unranked tree t is a binary tuple (D, L), where

- $D \subseteq \mathbb{N}^*$ is a finite *tree domain*, that is,
  - D is prefix closed,
  - if $xi \in D$, then $x0, \ldots, x(i-1) \in D$.
- $L : D \to \Sigma$.



U_Σ: The set of unranked trees over Σ.

Nondeterministic finite hedge automata (NFHA)

A *hedge*: A sequence of unranked trees $t_1, \ldots, t_n$.

A NFHA $\mathcal{A}$ is a tuple (Q, Σ, δ, F), where

- $F \subseteq Q$,
- δ is a finite set of transition rules of the form $a(R) \to q$, where
  - $a \in \Sigma$, $q \in Q$,
  - $R \subseteq Q^*$ is a regular language over Q.

  The languages R are called the *horizontal languages*.

A *run* of $\mathcal{A}$ over a tree $t = (D, L)$ is a tree $r_{\mathcal{A},t} = (D, L')$ s.t.

$\forall x \in D$ *labeled by* a *and with children* $x0, \ldots, xi$,
$\exists$ *a rule* $a(R) \to q$ *s.t.* $r_{\mathcal{A},t}(x0) \ldots r_{\mathcal{A},t}(xi) \in R$ *and* $r_{\mathcal{A},t} = q$.

A run $r_{\mathcal{A},t}$ is *accepting* if $r_{\mathcal{A},t}(\varepsilon) \in F$.

Let $L \subseteq U_\Sigma$. Then L is regular if $\exists$ a NFHA $\mathcal{A}$ recognizing L.

Normalised and deterministic NFHA

A *normalised* NFHA is a NFHA $\mathcal{A} = (Q, \Sigma, \delta, F)$ s.t.

$\forall (a, q) \in \Sigma \times Q$, $\exists$ *at most one rule of the form* $a(R) \to q$ *in* δ.

Proposition. $\forall$ NFHA $\mathcal{A}$, $\exists$ an equivalent normalised NFHA $\mathcal{A}'$.

Proof.

$a(R_1) \to q, a(R_2) \to q \Longrightarrow a(R_1 \cup R_2) \to q$. □

Normalised and deterministic NFHA

A *deterministic* finite hedge automaton (DFHA) is a NFHA $\mathcal{A} = (Q, \Sigma, \delta, F)$ s.t.

$$\forall a \in \Sigma,\ \textit{if}\ a(R_1) \to q_1, a(R_2) \to q_2\ \textit{s.t.}\ q_1 \neq q_2,\ \textit{then}\ R_1 \cap R_2 = \varnothing.$$

Proposition. $\forall$ NFHA $\mathcal{A}$, $\exists$ an equivalent DFHA $\mathcal{A}'$ of exponential size.

Proof.

Let $\mathcal{A} = (Q, \Sigma, \delta, F)$ be a NFHA.
Assume $\mathcal{A}$ is normalised. For $(a, q) \in \Sigma \times Q$, let $R_{a,q}$ denote $R : a(R) \to q$.
Define $\mathcal{A}' = (2^Q, \Sigma, \delta', F')$ as follows.

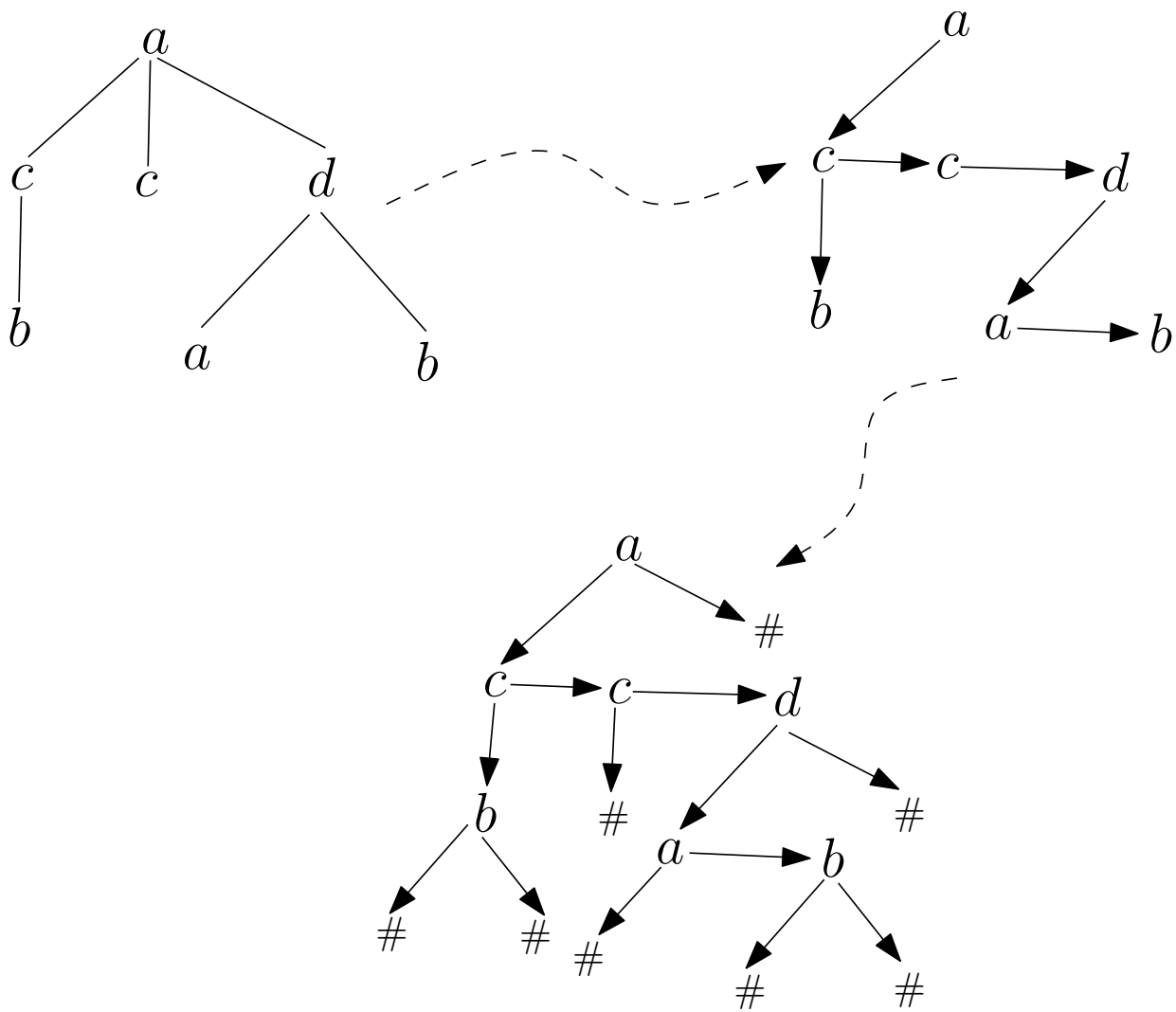
- $F' = \{S \mid S \cap F \neq \varnothing\}$,
- $\forall a \in \Sigma, S \in 2^Q$, $a(R') \to S \in \delta'$, where
  $S_1 \dots S_n \in R'$ iff $S = \{q | \exists q_1 \in S_1, \dots, q_n \in S_n.\ q_1 \dots q_n \in R_{a,q}\}$.
  R' is regular:
  Define $L_{a,q}$ as $\{S_1 \dots S_n \mid \exists q_1 \in S_1, \dots, q_n \in S_n.\ q_1 \dots q_n \in R_{a,q}\}$.
  Then $R' = (\bigcup_{q \in S} L_{a,q}) \backslash (\bigcup_{q \notin S} L_{a,q})$.

Outline

1. Hedge automata
2. Closure properties
3. Minimization
4. Decision problems

First-child-next-sibling (FCNS) encoding

An example:



First-child-next-sibling (FCNS) encoding

For every unranked tree t over the alphabet Σ, construct a ranked tree $\mathrm{fcns}(t)$ as follows.

- $\mathrm{fcns}(a) = a(\#, \#)$,
- $\mathrm{fcns}(a(t_1, \ldots, t_n)) = a(\mathrm{fcns}(t_1, \ldots, t_n), \#)$,
- for every hedge $t_1, \ldots, t_n$ with $n \geqslant 2$,
  $\mathrm{fcns}(t_1, \ldots, t_n) = \mathrm{fcns}(t_1) \mathbin{@} \mathrm{fcns}(t_2, \ldots, t_n)$, where $a(s, \#) \mathbin{@} s' = a(s, s')$.

For a tree language $L \subseteq U_\Sigma$, let $\mathrm{fcns}(L) = \{\mathrm{fcns}(t) \mid t \in L\}$.

Example:

$$\begin{aligned}\mathrm{fcns}(d(a,b)) &= d(\mathrm{fcns}(a,b), \#) = d(\mathrm{fcns}(a)@\mathrm{fcns}(b), \#) \\ &= d(a(\#,\#)@b(\#,\#), \#) = d(a(\#, b(\#,\#)), \#)\end{aligned}$$

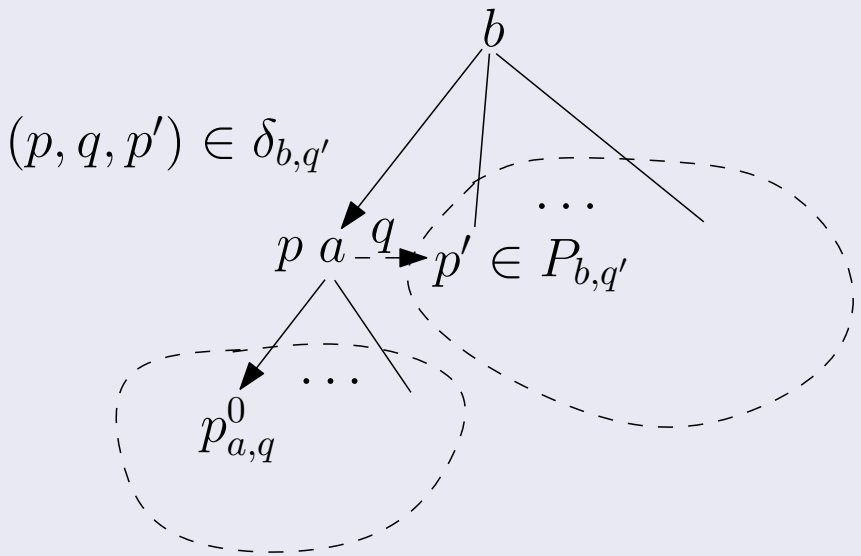
From unranked regular to ranked regular

Proposition. Let $L \subseteq U_\Sigma$. Then L is regular implies that fcns(L) is regular.

Proof.

Suppose L is recognized by a normalised NFHA $\mathcal{A} = (Q, \Sigma, \delta, F)$.
For every (a, q) s.t. $R_{a,q} \neq \emptyset$,
suppose that $R_{a,q}$ is recognized by a NFA $\mathcal{B}_{a,q} = (P_{a,q}, Q, \delta, p^0_{a,q}, F_{a,q})$.
W.l.o.g. assume that the state sets $P_{a,q}$ are disjoint from each other.

The intuition:



From unranked regular to ranked regular

Proposition. Let $L \subseteq U_\Sigma$. Then L is regular implies that fcns(L) is regular.

Proof.

Suppose L is recognized by a normalised NFHA $\mathcal{A} = (Q, \Sigma, \delta, F)$.
For every (a, q) s.t. $R_{a,q} \neq \emptyset$,
suppose that $R_{a,q}$ is recognized by a NFA $\mathcal{B}_{a,q} = (P_{a,q}, Q, \delta, p^0_{a,q}, F_{a,q})$.
W.l.o.g. assume that the state sets $P_{a,q}$ are disjoint from each other.

Define $\mathcal{A}' = (Q', \Sigma, \delta', F')$ as follows.

- $Q' = Q \cup \bigcup_{a,q} P_{a,q} \cup \{q_\#\}$,
- $F' = F$,
- δ' is defined by the following rules,
  - $(\#, q_\#) \in \delta'$,
  - $(p^0_{a,q}, q_\#, a, q) \in \delta'$,
  - if $\exists b, q, p', q'$ s.t. $p, p' \in P_{b,q'}$, $(p, q, p') \in \delta_{b,q'}$, then $(p^0_{a,q}, p', a, p) \in \delta'$,
  - if $\exists b, q, p', q'$ s.t. $\varepsilon \in R_{a,q}$, $p \in P_{b,q'}$, $p' \in F_{b,q'}$, $(p, q, p') \in \delta_{b,q'}$, then $(q_\#, q_\#, a, p) \in \delta'$,
  - if $\exists b, q, p', q'$ s.t. $\varepsilon \in R_{a,q}$, $p, p' \in P_{b,q'}$, $(p, q, p') \in \delta_{b,q'}$, then $(q_\#, p', a, p) \in \delta'$.

From ranked regular to unranked regular

Proposition. Let $L \subseteq T_{\Sigma \cup \{\#\}}$. Then L is regular implies $\mathrm{fcns}^{-1}(L)$ is regular.

A notation: $\mathrm{fcns}^{-1}(L) = \{t \in U_\Sigma \mid \mathrm{fcns}(t) \in L\}$.

Proof sketch.

Let $\mathcal{A} = (Q, \Sigma \cup \{\#\}, \delta, F)$ be a NBUT (over ranked trees).
We can define a NFHA $\mathcal{A}'$ such that the horizontal languages of $\mathcal{A}'$ are used to simulate the partial runs of $\mathcal{A}$ over the paths $x01^*$. □

Homework. Give the detailed construction for $\mathrm{fcns}^{-1}(L)$ from $\mathcal{A}$ in the above proof.

Closure properties

Corollary. The set of regular languages over unranked trees are closed under all Boolean operations.

Fact: The following facts hold for fcns and fcns^{-1}.

- $\text{fcns} : U_\Sigma \to T_\Sigma$ is an injective (non-surjective) function,
- $\text{fcns}^{-1} : T_\Sigma \to U_\Sigma$ is an injective and surjective partial function.

Proof.

Suppose $L_1, L_2 \subseteq U_\Sigma$ are regular.
The corollary follows from the closure property of regular languages over ranked trees and the following equations.

- $L_1 \cup L_2 = \text{fcns}^{-1}(\text{fcns}(L_1) \cup \text{fcns}(L_2))$, similarly for $\cap$,
- $U_\Sigma \backslash L_1 = \text{fcns}^{-1}(T_{\Sigma \cup \{\#\}} \backslash \text{fcns}(L_1))$.

Outline

1. Hedge automata
2. Closure properties
3. Minimization
4. Decision problems

Representation of horizontal languages

We use DFA or NFA to represent the horizontal languages,

- $NFHA(NFA)$: NFHA with horizontal languages represented by NFA,
- $NFHA(DFA)$: NFHA with horizontal languages represented by DFA,
- $DFHA(NFA)$: DFHA with horizontal languages represented by NFA,
- $DFHA(DFA)$: DFHA with horizontal languages represented by DFA.

Nondeterminism of DFHA(DFA)

Let $\mathcal{A} = (Q, \Sigma, \delta, F)$ be a DFHA(DFA).
W.l.o.g, assume that $\mathcal{A}$ is normalised:

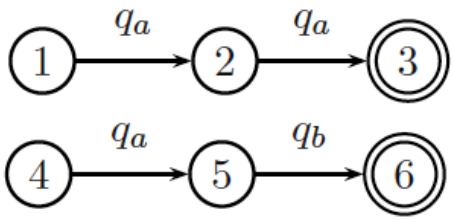
$\forall(a, q)$, $R_{a,q}$ is given by a DFA $\mathcal{B}_{a,q} = (Q_{a,q}, \Sigma, \delta_{a,q}, q^0_{a,q}, F_{a,q})$.

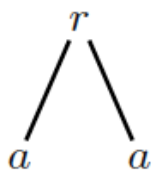
Size of a DFHA(DFA) $\mathcal{A}$: $|Q| + \sum_{a,q} |Q_{a,q}|$.

Nondet. choice of different DFAs:

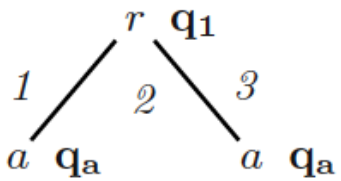
$\mathcal{A} = (\{q_1, q_2, q_a, q_b\}, \{r, a, b\}, \delta, \{q_1, q_2\})$, where δ is defined as follows,

- $a(\{\varepsilon\}) \to q_a$, $b(\{\varepsilon\}) \to q_b$,
- $r(\{q_a q_a\}) \to q_1$, $r(\{q_a q_b\}) \to q_2$.

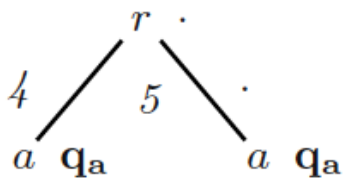




(a)



(b)

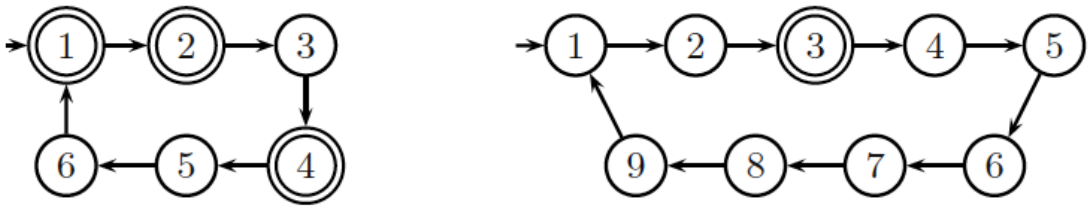


(c)

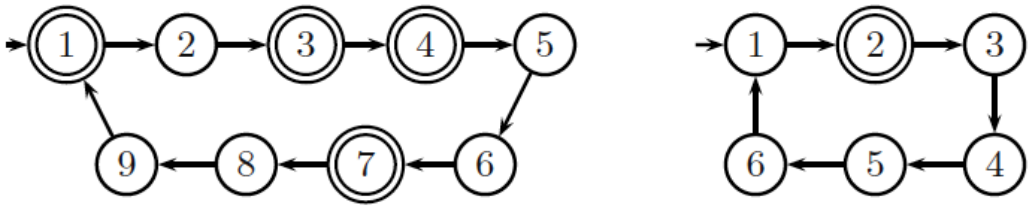
Minimization of DFHA(DFA) is dificult

No unique minimum DFHA(DFA) for a given language:

The language $\{r(a(w)) \mid w \in L\}$, where
$L = L_1 \cup L_2 \cup L_3$, $L_1 = (bbb)^*$, $L_2 = b(bbbbbb)^*$ and $L_3 = bb(bbbbbbbbb)^*$.



(a)



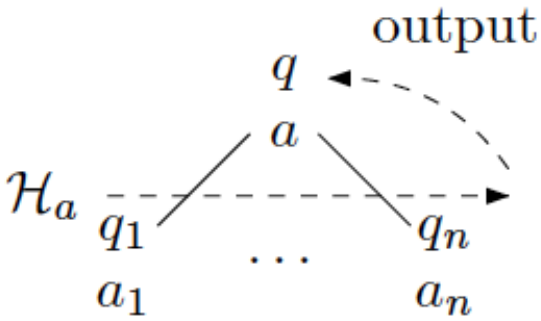
(b)

Theorem. DFHA(DFA) minimization is NP-complete.

Reference. W. Martens, and J. Niehren, Minimizing Tree Automata for Unranked Trees, DBPL 2005.

Deterministic stepwise hedge automata (DSHA)

The intuition (Deterministic horizontal automata with output):



A DSHA is a tuple $\mathcal{A} = (Q, \Sigma, \delta_0, F, \delta)$, where

- Q, Σ, F are as usual,
- $\delta_0 : \Sigma \to Q$ is the initial state assignment function,
- and $\delta : Q \times Q \to Q$ is the transition function.

For every $a \in \Sigma$, define δ_a^* as follows,

$$\delta_a^*(\varepsilon) = \delta_0(a),\ \delta_a^*(wq) = \delta(\delta_a^*(w), q).$$

A run of a DSHA $\mathcal{A} = (Q, \Sigma, \delta_0, F, \delta)$ over a tree $t = (D, L)$:

A tree $r_{\mathcal{A},t} = (D, L')$ s.t.

$\forall$ *node* x *with label* a *and children* $x0, \ldots, xk$,

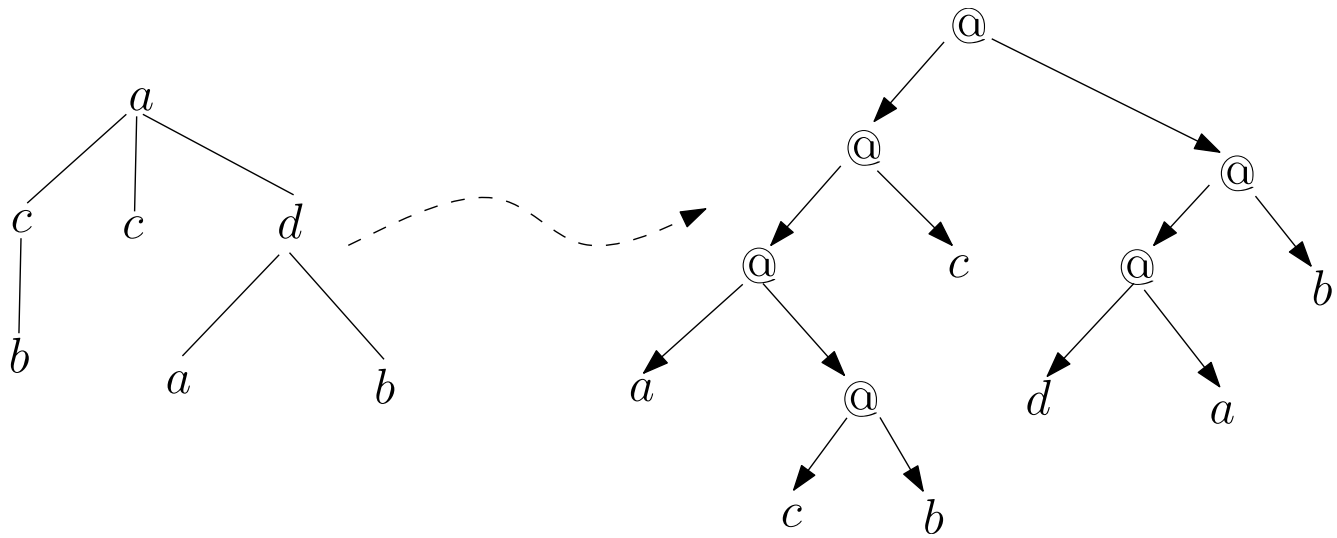
$$L'(x) = \delta_a^*(L'(x0) \ldots L'(xk)).$$

Extension operator

Extension operator @:

Given two trees $t, t' \in U_\Sigma$ s.t. $t = a(t_1, \ldots, t_n)$, then $t@t' = a(t_1, \ldots, t_n, t')$.

Encoding of unranked trees by extension operator:



The *extension encoding* $ext : U_\Sigma \to T_{\{@\}\cup\Sigma}$,

- $ext(a) = a$,
- $ext(a(t_1, \ldots, t_n)) = @(ext(a(t_1, \ldots, t_{n-1})), ext(t_n))$.

Proposition. $ext : U_\Sigma \to T_{\{@\}\cup\Sigma}$ is a bijection.

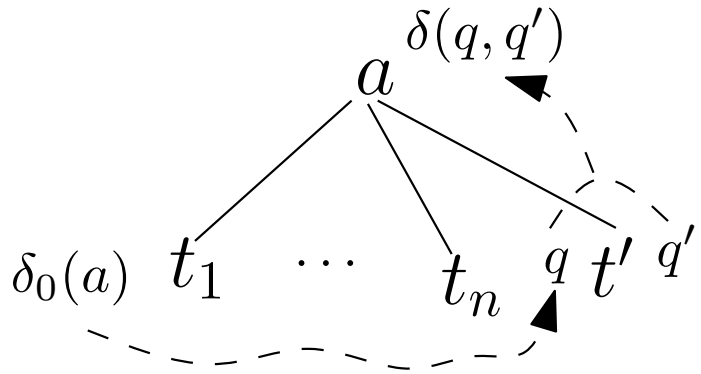
For $L \subseteq U_\Sigma$, define $ext(L) = \{ext(t) \mid t \in L\}$.

DSHA as DBUTA over $T_{\{@\}\cup\Sigma}$

The intuition:

ranked	stepwise
(a, q)	$\delta_0(a) = q$
$(q_1, q_2, @, q)$	$\delta(q_1, q_2) = q$

Lemma. Let $\mathcal{A} = (Q, \Sigma, \delta_0, F, \delta)$ be a DSHA and $t, t' \in U_\Sigma$ s.t. $r_{\mathcal{A},t}(\varepsilon) = q$ and $r_{\mathcal{A},t'}(\varepsilon) = q'$, then $r_{\mathcal{A},t@t'}(\varepsilon) = \delta(q, q')$.



DSHA as DBUTA over $T_{\{@\}\cup\Sigma}$

The intuition:

ranked	stepwise
(a, q)	$\delta_0(a) = q$
$(q_1, q_2, @, q)$	$\delta(q_1, q_2) = q$

Lemma. Let $\mathcal{A} = (Q, \Sigma, \delta_0, F, \delta)$ be a DSHA and $t, t' \in U_\Sigma$ s.t. $r_{\mathcal{A},t}(\varepsilon) = q$ and $r_{\mathcal{A},t'}(\varepsilon) = q'$, then $r_{\mathcal{A},t@t'}(\varepsilon) = \delta(q, q')$.

For a DSHA $\mathcal{A} = (Q, \Sigma, \delta_0, F, \delta)$, define $ext(\mathcal{A}) = (Q, \{@\} \cup \Sigma, \delta', F)$, where

- for every $a \in \Sigma$, $(a, q) \in \delta'$ iff $\delta_0(a) = q$,
- $(q_1, q_2, @, q) \in \delta'$ iff $\delta(q_1, q_2) = q$.

Proposition. For every DSHA $\mathcal{A}$, $L(ext(\mathcal{A})) = ext(L(\mathcal{A}))$.

Minimization of DSHA

Theorem. For each regular language $L \subseteq U_\Sigma$, there is a unique minimum DSHA recognizing L.

Proof.

Suppose $\mathcal{A}_1$ and $\mathcal{A}_2$ are two minimum DSHAs for L.
Then $ext(\mathcal{A}_1)$ and $ext(\mathcal{A}_2)$ are two minimum DBTUAs for $ext(L)$.
Therefore, $ext(\mathcal{A}_1) \cong ext(\mathcal{A}_2)$, so $\mathcal{A}_1 \cong \mathcal{A}_2$. □

Minimization of DSHA

Theorem. For each regular language $L \subseteq U_\Sigma$, there is a unique minimum DSHA recognizing L.

Define a congruence $\equiv_L$ (wrt. @) over U_Σ as follows:

$$t_1 \equiv_L t_2 \textit{ iff } ext(t_1) \equiv_{ext(L)} ext(t_2).$$

Theorem. Let $L \subseteq U_\Sigma$. Then L is regular iff $\equiv_L$ is of finite index.

Proof.

Suppose $\equiv_L$ is of finite index. Then L is recognized by the DSHA $(Q, \Sigma, \delta_0, F, \delta)$, where

- $Q = \{[t] \mid t \in U_\Sigma\}$, $F = \{[t] \mid t \in L\}$,
- $\delta_0(a) = [a]$, $\delta([t_1], [t_2]) = [t_1@t_2]$.

Suppose L is regular. Then $ext(L)$ is also regular.
So $\equiv_{ext(L)}$, and thus $\equiv_L$, is of finite index. □

Outline

1. Hedge automata
2. Closure properties
3. Minimization
4. Decision problems

Membership

Theorem. The membership problem for NFHA(NFA) is in PTIME.

Proof.

Let $\mathcal{A} = (Q, \Sigma, \delta, F)$ be a NFHA(NFA) and t be a tree.
The following problem can be solved in PTIME.

> *Given a NFA* $\mathcal{B} = (Q', Q, \delta', q_0, F')$, *a word* $S_1 \dots S_n \in (2^Q)^*$,
> *decide whether* $\exists q_1 \dots q_n$ *s.t.* $\forall i.\ q_i \in S_i$, *and* $q_1 \dots q_n$ *is accepted by* $\mathcal{B}$.

1. Compute the set of reachable states
   - $P_0 = \{q_0\}$,
   - for $i > 0$, $P_i = \{q' \mid \exists p' \in P_{i-1}, q \in S_i, (p', q, q') \in \delta'\}$.
2. Check whether $P_n \cap F' \neq \varnothing$.

Then the set of reachable states of $\mathcal{A}$ after reading t in bottom-up can be computed in PTIME. □

Emptiness

Theorem. The emptiness problem for NFHA(NFA) is in PTIME.

Proof.

Let $\mathcal{A} = (Q, \Sigma, \delta, F)$ be a normalised NFHA(NFA).

A state q is reachable if $\exists$ a run of $\mathcal{A}$ over a tree t, say r, s.t. $r(\varepsilon) = q$.

Compute the set of reachable states $R_{\mathcal{A}}$ as follows, until $R_{\mathcal{A},i} = R_{\mathcal{A},i-1}$.

- $R_{\mathcal{A},0} = \{q \mid \exists a, R.\ a(R) \to q \in \delta, \varepsilon \in R\}$,
- $R_{\mathcal{A},i} = R_{\mathcal{A},i-1} \cup \{q \mid \exists a, R.\ a(R) \to q, R \cap (R_{\mathcal{A},i-1})^* \neq \varnothing\}$.

Claim. $L(\mathcal{A})$ is nonempty iff $R_{\mathcal{A}} \cap F \neq \varnothing$. □

Inclusion

Theorem. The inclusion problem of NFHA(NFA) is EXPTIME-complete.

Proof.

Upper bound:
Let $\mathcal{A}, \mathcal{A}'$ be two NFHA(NFA).

1. Determinize $\mathcal{A}'$: In EXPTIME, obtaining a complete DFHA(DFA) $\mathcal{A}''$,
2. Complement $\mathcal{A}''$ to get $\mathcal{A}'''$: Just complementing the set of accepting states,
3. Decide whether $L(\mathcal{A}) \cap L(\mathcal{A}''')$ is empty: In polynomial time over the size of $\mathcal{A}$ and $\mathcal{A}'''$.

Lower bound:
Reduction from APSPACE TMs to the universality of NFHA(NFA).
Similar to the lower bound for the universality of NBTUA over ranked trees. □

Determinism

Theorem. Checking whether a NFHA(NFA) is deterministic is in PTIME.

Proof.

Let $\mathcal{A} = (Q, \Sigma, \delta, F)$ be a normalised NFHA(NFA) s.t. every $R_{a,q}$ is given by a NFA $\mathcal{B}_{a,q}$.
The following computation is in PTIME:

$$\forall a(R_{a,q_1}) \to q_1, a(R_{a,q_2}) \to q_2 \in \delta \ s.t. \ q_1 \neq q_2, \ test \ R_{a,q_1} \cap R_{a,q_2} = \varnothing.$$

Completeness

A *complete* NFHA is a NFHA $\mathcal{A} = (Q, \Sigma, \delta, F)$ s.t.

$$\forall t,\ \exists \textit{ at least one run of } \mathcal{A} \textit{ over } t.$$

Theorem. Completeness of NFHA(NFA) (resp. DFHA(DFA)) is PSPACE-c.

Proof.

PSPACE-hardness: Reduction from intersection of DFAs.
Suppose $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are n DFAs s.t. $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, q_{0,i}, F_i)$.
Construct a DFHA(DFA) $\mathcal{B} = (Q', \Sigma', \delta', F')$, where

- $Q' = \{q_1, q_r\} \cup \{q_a \mid a \in \Sigma\}$, $\Sigma' = \Sigma \cup \{r\}$, $F' = \{q_r\}$,
- δ' is defined as follows.
  Let $R_i = prj(L(\overline{\mathcal{A}_i}))$ (where $prj(a) = q_a$), $R' = Q'^*(\{q_r, q_1\})Q'^*$.
  Then $\delta' = \{a(R_i) \to q_a, a(R') \to q_1, r(R_i) \to q_r, r(R') \to q_1\}$.

Claim.
$\mathcal{B}$ is complete iff $\bigcup_i R_i = (\{q_a \mid a \in \Sigma\})^*$ iff $\bigcup_i L(\overline{\mathcal{A}_i}) = \Sigma^*$ iff $\bigcap_i L(\mathcal{A}_i) = \varnothing$.

Completeness

A *complete* NFHA is a NFHA $\mathcal{A} = (Q, \Sigma, \delta, F)$ s.t.

$\forall t$, $\exists$ *at least one run of* $\mathcal{A}$ *over* t.

Theorem. Completeness of NFHA(NFA) (resp. DFHA(DFA)) is PSPACE-c.

Proof.

In PSPACE:

Let $\mathcal{A} = (Q, \Sigma, \delta, F)$ be a normalised NFHA(NFA)

s.t. the languages $R_{a,q}$ *are defined by NFAs* $\mathcal{B}_{a,q}$.

W.l.o.g assume that all the states of $\mathcal{A}$ are reachable.

A state q *is reachable if* $\exists$ *a run of* $\mathcal{A}$ *over a tree* t, *say* r, *s.t.* $r(\varepsilon) = q$.

Claim. $\mathcal{A}$ is complete iff $\forall a \in \Sigma. \bigcup_{q \in Q} R_{a,q} = Q^*$.

From the NFAs $\mathcal{B}_{a,q}$,

- an NFA $\mathcal{C}_a$ can be constructed in PTIME to recognize $\bigcup_{q \in Q} R_{a,q}$,
- the universality of $\mathcal{C}_a$ can be checked in PSPACE.

□

(Possibly) Open questions

DFHA(DFA):

The complexity of inclusion problem

Next lecture

Applications to model checking