Nondeterministic Tree Width of Regular Languages

Cezar Câmpeanu* Kai Salomaa†

25.06.2015

* Department of Computer Science
University of Prince Edward Island
CANADA
† School of Computer Science
Queen's University, CANADA

Measures of Non-Determinism

Nondeterminism

- Nondeterminism plays a fundamental role in the theory of computation.
- For some machine models, nondeterminism enhances the computational power of the model (pushdown automata), while for others it does not (Turing machines, finite automata).
- For resource bounded Turing machines, the relationship between determinism and nondeterminism leads to very difficult open problems (P vs. NP).
- Finite automata operate in real time, and the "resource" to measure is the number of states (*descriptional complexity*)
  - ❖ . . . we can measure also nondeterminism and consider ***trade-offs*** between (the amount of) nondeterminism and size of the machine

Nondeterminism

- Nondeterminism plays a fundamental role in the theory of computation.
- For some machine models, nondeterminism enhances the computational power of the model (pushdown automata), while for others it does not (Turing machines, finite automata).
- For resource bounded Turing machines, the relationship between determinism and nondeterminism leads to very difficult open problems (P vs. NP).
- Finite automata operate in real time, and the "resource" to measure is the number of states (*descriptional complexity*)
    - ❖ . . . we can measure also nondeterminism and consider ***trade-offs*** between (the amount of) nondeterminism and size of the machine

Nondeterminism

- Nondeterminism plays a fundamental role in the theory of computation.
- For some machine models, nondeterminism enhances the computational power of the model (pushdown automata), while for others it does not (Turing machines, finite automata).
- For resource bounded Turing machines, the relationship between determinism and nondeterminism leads to very difficult open problems (P vs. NP).
- Finite automata operate in real time, and the "resource" to measure is the number of states (*descriptional complexity*)
  - ❖ ...we can measure also nondeterminism and consider ***trade-offs*** between (the amount of) nondeterminism and size of the machine

Measures for NFAs

- We measure the "amount of nondeterminism" of a ***finite automaton***.
- Several approaches considered in the literature:
  - the number of accepting configurations for a given input (degree of ambiguity)
  - the number of partial computations for a given input (tree width)
  - the amount of nondeterminism on a single best (or worst) computation on a given input (branching)
  - the size of look-ahead (guessing measure)
  - the number of non-deterministic choices in a computation (advice measure)

Measures for Languages

- In case a minimal NFA for a language L is a DFA, we only have one possible computation, thus all the measures considered before are equal to 1 (including the tree width).
- For a language L we may have two different (even minimal) NFA's, such that the tree width of a computation on a given input is quite different.
- The nondeterministic width of a language L is defined as the least tree width of any state-minimal NFA recognizing L.

$$\mathrm{tw}(L) = \inf\{\mathrm{tw}(A) \mid L(A) = L, A \text{ is a minimal } NFA\}. \tag{1}$$

Tree width

- The tree width of A on w, $\mathrm{tw}_A(w)$, is the number of partial computations of A on w.

$$\mathrm{tw}(A) = \sup\{\mathrm{tw}_A(w) \mid w \in \Sigma^*\}.$$

- A has finite tree width if $\mathrm{tw}(A)$ is finite.
- (A. Palioudakis, et all JALC 2012) An NFA A has finite tree width if and only if no cycle of A contains a nondeterministic transition.
- $$\mathrm{nsc}_{\mathrm{tw}\le k}(L) = \inf\{\mathrm{size}(A) \mid A \text{ is an } NFA, L = L(A), \text{ and } \mathrm{tw}(A) \le k\}. \quad (2)$$
- The tree width of a regular language L is

$$\mathrm{tw}(L) = \inf\{\mathrm{tw}(A) \mid L(A) = L, A \text{ is a minimal } NFA\}. \quad (3)$$

Tree width of an NFA

- Tree width — not related to graph theory “tree width”
- Tree width counts the number of paths in computation trees of an NFA.
- This notion is called “leaf size” by Hromkovič et al. (2002) or “computations(A)” by Björklund and Martens (2012)

Example

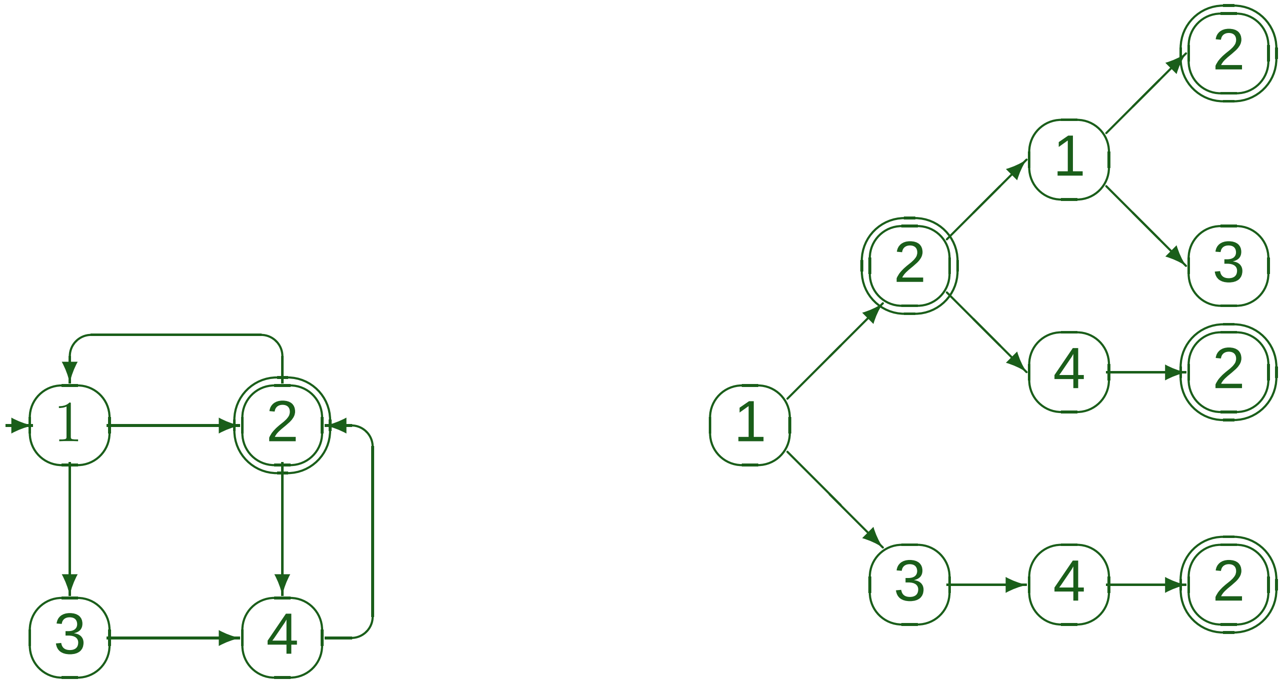


Figure 1: A unary NFA A and its computation tree on input a^3.

The *tree width* of A on input a^3 is four, $\mathrm{tw}_A(a^3) = 4$.

Infinite vs Finite Tree Width

- If language L has infinite tree width, this means that all minimal NFAs for L must have unbounded tree width.
- If L has tree width one, then the unique minimal DFA for L is also minimal as an NFA.
- $\mathrm{tw}(\Sigma^* w) = 1$.
- $\mathrm{tw}(L_k) = \infty$, where $L_k = \Sigma^* b \Sigma^{k-1}$, $k \geq 2$.
- $\mathrm{tw}(L_{a,k}) = \infty$, where $L_k = \Sigma^* (\Sigma - a) \Sigma^{k-1}$, $k \geq 2$.

Computing Tree Width

- Computing the tree width of a regular language is PSPACE-complete.
- The problem of deciding for a given NFA A with tree width k, and for a given $m \leq k$ whether or not $\mathrm{tw}(L(A)) = m$ is in coNP.
- Deciding if $\mathrm{tw}(L) = 1$ is NP hard even for unary languages.
- the proof uses a modification of the well-known hardness proof for the union-universe problem for DFAs: for a polynomial space bounded TM M and input string x, we construct an NFA D (having size polynomial in $|x|$), where D accepts the set of strings that are not accepting computations of M on x. Then $L(D)$ has tree width one iff M does not accept x.

Operations

Union

- $L_k = \bigcup_{w \in \Sigma^{k-1}} L_w$.
  $\mathrm{tw}(L_k) = \infty$ and $\mathrm{tw}(L_w) = 1$.
- There must be R_1, R_2 such that $\mathrm{tw}(R_1) < \infty$, $\mathrm{tw}(R_2) < \infty$ and $\mathrm{tw}(R_1 \cup R_2) = \infty$.
- $L_1 = L((a+b)^*baaa)$, $L_2 = L((a+b)^*baba)$
  $L_1 \cup L_2 = L((a+b)^*ba(a+b)a)$
  $\mathrm{tw}(L_i) = 1$, $i = 1, 2$, but $\mathrm{tw}(L_1 \cup L_2) = \infty$

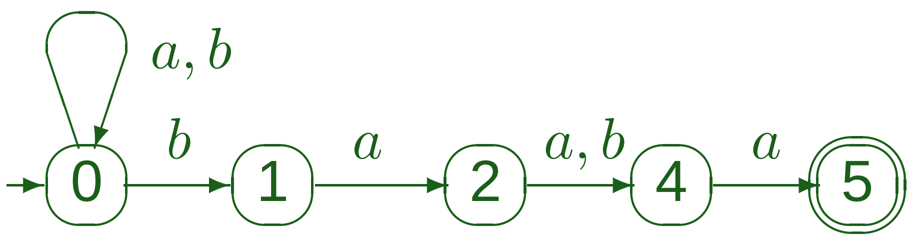


Figure 2: A minimal NFA for the union of $L((a+b)^*baba)$ and $L((a+b)^*baaa)$

Concatenation, Reversal, and Complement

- Concatenation: $L_1 = \Sigma^*$, $L_2 = b\Sigma^{k-1}$, $\mathrm{tw}(L_1) = 1$, $\mathrm{tw}(L_2) = 1$, $\mathrm{tw}(L_1 \cup L_2) = \mathrm{tw}(L_k) = \infty$.
- Reversal $\mathrm{tw}(L_k^R) = 1$.
- Complement
  $L = \{\varepsilon, a, a^2, a^4\}$, $tw(L) = 1$
  $\overline{L} = L((a^2)^*(a^3)^+)$.
  Minimal NFA for $\overline{L}$ has 5 states and any finite tree width for L needs at least 6 states, cf. Palioudakis, Salomaa, Akl: Proceedings of SOFSEM 2014.

Intersection

- $L_1 = L((a^*)) \setminus \{\varepsilon, a, a^2\}$ and $L_2 = L((a^*)) \setminus \{\varepsilon, a^2, a^4\}$. $L_1 \cap L_2 = L((a^*)) \setminus \{\varepsilon, a, a^2, a^4\}$, thus $\mathrm{tw}(L_1 \cap L_2) = \infty$ and $\mathrm{tw}(L_i) = 1$, $i = 1, 2$

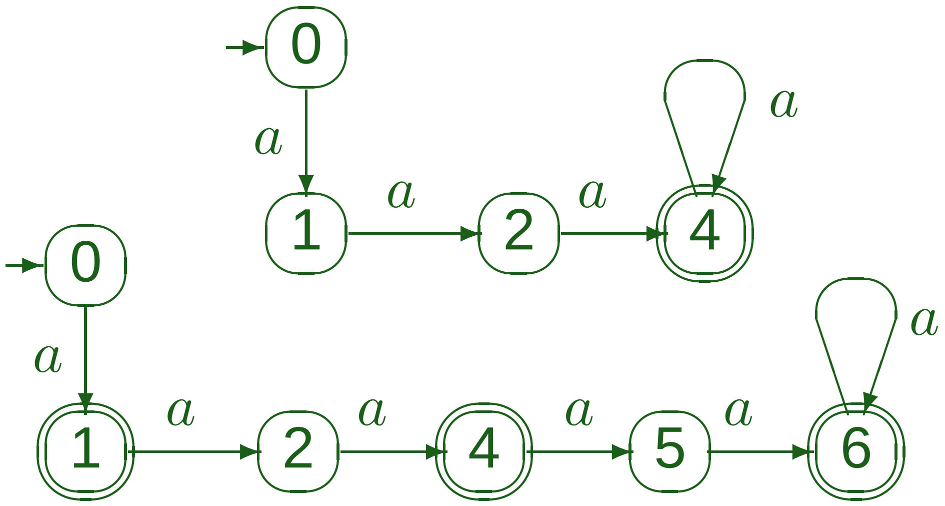


Figure 3: A minimal NFA for L_1, up, and L_2, down.

Conclusion

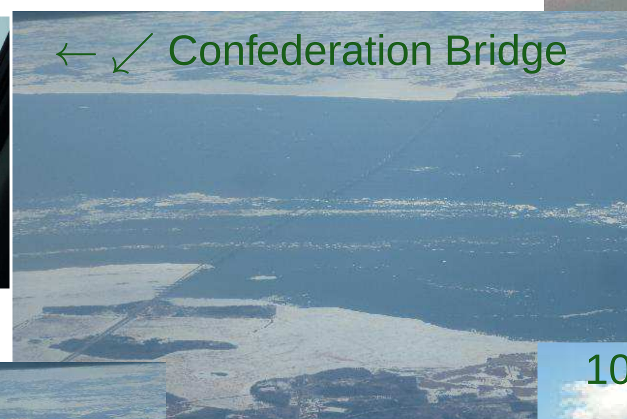
Conclusions and Open Problems

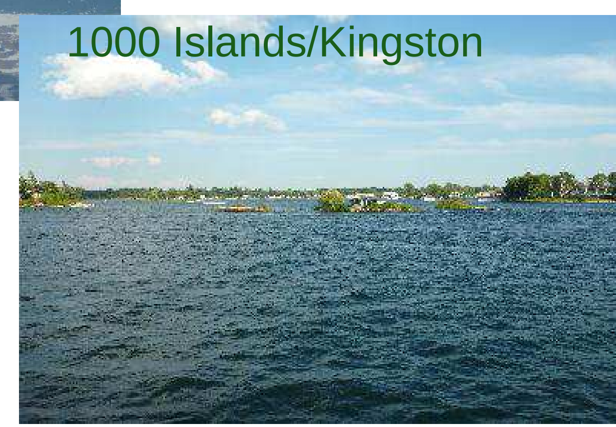
- Computing tree width of a language is hard.
- If L_1 and L_2 are regular languages with finite tree width, for most operations $\circ$, there is no upper-bound for $\mathrm{tw}(L_1 \circ L_2)$.
  - ❖ Find an interesting operation for which we can find an upper-bound.
- Some examples are closely related to examples for deterministic regular expressions.
  - ❖ Find some strong relation between regular expression ambiguity and measures of non-determinism induced by NFAs.


Queens University


UPEI


PEI


← ↙ Confederation Bridge


1000 Islands/Kingston

Thank You!