Regular Expressions and NFAs without ε-Transitions

Georg Schnitger[⋆]

Institut für Informatik, Johann Wolfgang Goethe-Universität,
Robert Mayer Straße 11–15, 60054 Frankfurt am Main, Germany
georg@thi.informatik.uni-frankfurt.de

Abstract. We consider the problem of converting regular expressions into ε-free NFAs with as few transitions as possible. If the regular expression has length n and is defined over an alphabet of size k, then the previously best construction uses $O(n \cdot \min\{k, \log_2 n\} \cdot \log_2 n)$ transitions. We show that $O(n \cdot \log_2 2k \cdot \log_2 n)$ transitions suffice. For small alphabets, for instance if $k = O(\log_2 \log_2 n)$, we further improve the upper bound to $O(k^{1+\log^* n} \cdot n)$. In particular, $O(2^{\log_2^* n} \cdot n)$ transitions and hence almost linear size suffice for the binary alphabet! Finally we show the lower bound $\Omega(n \cdot \log_2^2 2k)$ and as a consequence the upper bound $O(n \cdot \log_2^2 n)$ of [7] for general alphabets is best possible. Thus the conversion problem is solved for large alphabets ($k = n^{\Omega(1)}$) and almost solved for small alphabets ($k = O(1)$).

Keywords: Automata and formal languages, descriptional complexity, nondeterministic automata, regular expressions.

1 Introduction

One of the central tasks on the border between formal language theory and complexity theory is to describe infinite objects such as languages by finite formalisms such as automata, grammars, expressions among others and to investigate the descriptional complexity and capability of these formalisms. Formalisms like expressions and finite automata have proven to be very useful in building compilers, and techniques converting one formalism into another were used as basic tools in the design of computer systems such as UNIX ([12] and [5], p. 123).

A typical application in lexicographical analysis starts with a regular expression that has to be converted into an ε-free nondeterministic finite automaton. Here, the descriptional complexity of an expression R is its length and the descriptional complexity of a nondeterministic finite automaton (NFA) is the number of its edges or transitions, where identical edges with distinct labels are differentiated.

All classical conversions [1, 3, 9, 12] produce ε-free NFAs with worst-case size quadratic in the length of the given regular expression and for some time this was assumed to be optimal [10]. But then Hromkovic, Seibert and Wilke [7] constructed ε-free NFAs with surprisingly only $O(n(\log_2 n)^2)$ transitions for regular expressions of length n and this transformation can even be implemented to run in time $O(n \cdot \log_2 n + m)$, where m is the size of the output [4]. Subsequently Geffert [2] showed that even ε-free NFAs with $O(n \cdot k \cdot \log_2 n)$ transitions suffice for alphabets of size k, improving the bound of [7] for small alphabets.

We considerably improve the upper bound of [2] for alphabets of small size k. In particular we show

[⋆] Work supported by DFG grant SCHN 503/4-1.

Theorem 1. *Every regular expression R of length n over an alphabet of size k can be recognized by an ε-free NFA with at most*

$$O(n \cdot \min\{\log_2 n \cdot \log_2 2k, k^{1+\log^* n}\})$$

transitions.

As a first consequence we obtain ε-free NFAs of size $O(n \cdot \log_2 n \cdot \log_2 2k)$ for regular expressions of length n over an alphabet of size k. For small alphabets, for instance if $k = O(\log_2 \log_2 n)$, the upper bound $O(n \cdot k^{1+\log^* n})$ is better. In particular, $O(n \cdot 2^{\log_2^* n})$ transitions and hence almost linear size suffice for the binary alphabet.

A first lower bound was also given in [7], where it is shown that the regular expression

$$E_n = (1 + \varepsilon) \cdot (2 + \varepsilon) \cdots (n + \varepsilon)$$

over the alphabet $\{1, \ldots, n\}$ requires NFAs of size at least $\Omega(n \cdot \log_2 n)$. Lifshits [8] improves this bound to $\Omega(n(\log_2 n)^2 / \log_2 \log_2 n)$. We use ideas developed in [8] to prove the following optimal asymptotic bound for E_n.

Theorem 2. *There are regular expressions of length n over an alphabet of size k such that any equivalent ε-free NFA has at least $\Omega(n \cdot \log_2^2 2k)$ transitions.*

Thus the construction of [7] is optimal for large alphabets, i.e., if $k = n^{\Omega(1)}$. Since Theorem 1 is almost optimal for alphabets of fixed size, only improvements for alphabets of intermediate size, i.e., $\omega(1) = k = n^{o(1)}$, are still required.

In Section 2 we show how to construct small ε-free NFAs for a given regular expression R using ideas from [2, 7]. We obtain in Lemma 1 the upper bound $O(n \cdot \log_2 n \cdot \log_2 2k)$ by short-cutting ε-paths within the canonical NFA. Whereas the canonical NFA (with ε-transitions) is derived from the expression tree of R, the shortcuts are derived from a decomposition tree which is a balanced version of the expression tree. The subsequent improvement for small alphabets is based on repeatedly applying the previous upper bound to larger and larger subexpressions. We show the lower bound for the regular expression E_n in Section 3. Conclusions and open problems are stated in Section 4.

2 Small ε-Free NFAs for Regular Expressions

We first describe the canonical construction of an NFA (with ε-transitions), given the expression tree of a given regular expression R. We then proceed in the next section by defining a decomposition tree for R. After showing in section 2.2 how to obtain a small ε-free NFA from the decomposition tree we then recursively apply our recipe in section 2.3 to obtain close to optimal ε-free NFAs.

First observe that $R = R_1 \overset{+}{\circ} R_2$ or $R = S^*$ for subexpressions R_1, R_2, S of R. Fig. 1 shows this recursive expansion in NFA-notation. Thus, after completing this recursive expansion, we arrive at the "canonical" NFA N_R with a unique initial state q_0 and a unique final state q_f. Moreover, no transition enters q_0 and no transition leaves q_f; thus q_f is a trap state. Finally observe that N_R has at most $O(n)$ transitions for any regular expression R of length n.

2.1 Decomposition Trees

Let T_R be the **expression tree** of R with root r. To define decomposition trees we first introduce partial cuts, where a partial cut is a set of nodes of T_R with no two nodes of C being ancestors or descendants of each other. We define $T_R^v(C)$, for a

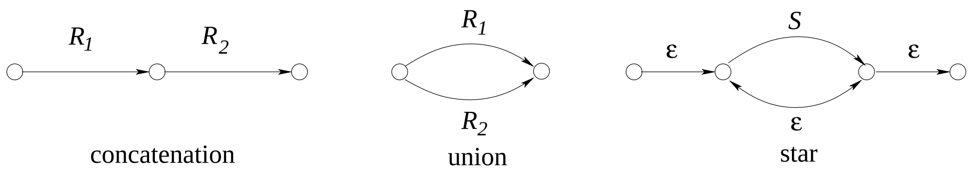


Fig. 1. The initial step in determining the NFA N_R for a regular expression R. The undirected version of N_R is a series-parallel graph with a unique source and unique sink. The sink is a trap state.

node v and a partial cut C of T_R, as the subtree of T_R with root v and all children-links for nodes in C removed. Hence nodes in C which are also descendants of v are (artificial) leaves in $T^v_R(C)$. Moreover we label the leaf for $x \in C$ with an artificial symbol denoting the regular subexpression determined by x in T_R: if $R^v(C)$ denotes the subexpression specified by $T^v_R(C)$, then $R^v(C)$ contains for each artificial leaf the corresponding artificial symbol. Finally $N^v_R(C)$ is the NFA obtained by recursively expanding $R^v(C)$ except for the artificial symbols of $R^v(C)$; any artificial symbol S is modeled by an artificial transition $p \xrightarrow{S} q$, where p is the unique initial state and q is the unique final state of the canonical NFA for S.

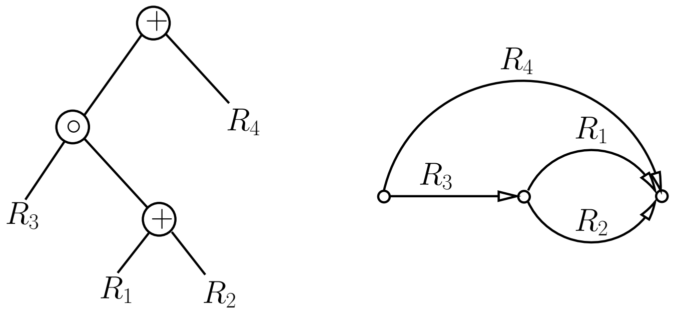


Fig. 2. Expression tree and canonical NFA with artificial transitions for $R = (R_3 \circ (R_1 + R_2)) + R_4$

We introduce the **decomposition tree** T^*_R for R as a balanced, small depth version of T_R. We begin by determining a **separating node** v of T_R, namely a node of T_R with a subtree of at least $\frac{n}{3}$, but less than $\frac{2n}{3}$ leaves. Then $T_1 = T^v_R(\emptyset)$ is the subtree of T_R with root v and T_1 determines the regular expression $S = R^v(\emptyset)$. We remove the edge connecting v to its parent, reattach v as an artificial leaf labeled with the artificial symbol S and obtain the second subtree $T_2 = T^r_R(\{v\})$ specifying the regular expression $R^r(\{v\})$. We obtain the original expression R after replacing the artificial symbol S in $R^r(\{v\})$ by the expression $R^v(\emptyset)$. $N^r_R(\{v\})$ contains a unique transition $q_1 \xrightarrow{S} q_2$ with label S. We obtain N_R from $N^r_R(\{v\})$ after identifying the unique initial and final states of $N^v_R(\emptyset)$ with q_1 and q_2 respectively and then replacing the transition $q_1 \xrightarrow{S} q_2$ by $N^v_R(\emptyset)$.

To define the decomposition tree T^*_R we create a new root s. We say that $q_1 \xrightarrow{S} q_2$ is the **artificial transition of s** and label s with the quadruple $(r, \emptyset, q_1 \xrightarrow{S} q_2, v)$. In general, if we label a node t of T^*_R with $(u, C, q_1 \xrightarrow{S} q_2, v)$,

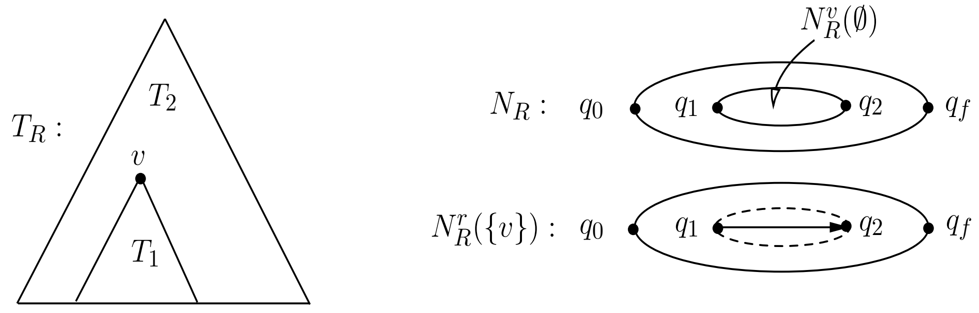


Fig. 3. The first expansion step for T^*_R and the corresponding NFAs.

- then we say that t represents the expression tree $T^u_R(C)$ as well as the NFA $N^u_R(C)$. When expanding node t, the NFA $N^u_R(C)$ is decomposed using the separating node v.
- The left child of t represents the expression tree $T^v_R(C)$ as well as the NFA $N^v_R(C)$ which has the unique initial state q_1 and the unique final state q_2.
- The right child of t represents the expression tree $T^u_R(C \cup \{v\})$ and the NFA $N^v_R(C \cup \{v\})$.
- One obtains the NFA $N^u_R(C)$ from the NFA $N^v_R(C \cup \{v\})$ of the right child after replacing the artificial transition $q_1 \xrightarrow{S} q_2$ by the NFA $N^v_R(C)$ of the left child.

We recursively repeat this expansion process for the left child of s representing $T^v_R(\emptyset)$ as well as the right child representing $T^r_R(\{v\})$. Observe that separating nodes have to have at least $\frac{N}{3}$, but less than $\frac{2N}{3}$ *original* (i.e., non-artificial) leaves in their subtrees, where N is the current number of original leaves.

We continue to expand until all trees contain at most one original leaf. If we have reached a node t of T^*_R whose expression tree $T^u_R(C)$ has exactly one leaf l (representing the original transition $q_1 \xrightarrow{a} q_2$), then we label t by the quadruple $(u, C, q_1 \xrightarrow{a} q_2, l)$ and stop the expansion. We summarize the important properties of T^*_R.

Proposition 1. *Let R be a regular expression of length n.*

(a) Each ε-free transition of N_R appears exactly once as an artificial transition of a leaf of T^*_R.

(b) Assume that node t is labeled with $(u, C, q_1 \xrightarrow{S} q_2, v)$. Then any ε-path in N_R from a state of the "left child NFA" $N^v_R(C)$ to a state outside of $N^v_R(C)$ has to traverse q_2. Any ε-path in N_R from a state outside of $N^v_R(C)$ to a state of $N^v_R(C)$ traverses q_1.

(c) Let $p \xrightarrow{a} q$ and $r \xrightarrow{b} s$ be two ε-free transitions of N_R corresponding to leaves l_1, resp. l_2 of T^*_R. Moreover let t be the lowest common ancestor of l_1 and l_2 in T^*_R. If there is an ε-path $q \xrightarrow{*} r$ in N_R and if $q_1 \xrightarrow{S} q_2$ is the artificial transition of t, then the path traverses q_1 or q_2.

(d) The depth of T^*_R is bounded by $O(\log_2 n)$.

Proof. (a) Any ε-free transition of N_R appears exactly once as a leaf of the expression tree T_R. The claim follows, since each expansion step for the decomposition tree T^*_R decomposes T_R.

(b) When "growing" the canonical NFA while expanding the expression tree T_R, we always replace an artificial transition by an NFA N with a unique initial and final state. No state outside of N will ever be linked directly with a state inside of

N. Moreover N can only be entered through its initial state and left through its final state.

(c) The lowest common ancestor t represents the expression $R^u(C)$ for some node u and a cut C. If w is the separating node of t, then $R^u(C)$ is decomposed into the expressions $R^w(C)$, recognized by $N_R^w(C)$, and $R^u(C \cup \{w\})$, recognized by $N_R^u(C \cup \{w\})$. Assume that both endpoints of, say, $p \xrightarrow{a} q$ belong to $N_R^w(C)$, the endpoints of $r \xrightarrow{b} s$ lie outside. But according to part (b), the left child NFA $N_R^w(C)$ can only be entered through its initial or final state which coincides with an endpoint of the artificial transition of t.

(d) follows, since the number of original leaves is reduced each time by at least the factor $\frac{2}{3}$. □

2.2 Constructing ε-Free NFAs from the Decomposition Tree

Property (c) of Proposition 1 is of particular importance when building a small ε-free NFA from the regular expression R, resp. from the NFA N_R: assume for instance that there are ε-paths $q_1 \xrightarrow{*} p_2$, $q_2 \xrightarrow{*} p_3$ and $q_3 \xrightarrow{*} p_4$ as well as ε-free transitions $p_i \xrightarrow{a_i} q_i$ for $i = 1, \ldots, 4$ in N_R. Then there is a path P from p_1 to q_4 built from the ε-transition and the four ε-free transitions. How can we simulate P without ε-transitions?

Assume that within the decomposition tree u is the lowest common ancestor (lca) of $p_1 \xrightarrow{a_1} q_1$ and $p_2 \xrightarrow{a_2} q_2$, v the lca of $p_2 \xrightarrow{a_2} q_2$ and $p_3 \xrightarrow{a_3} q_3$ and finally that w is the lca of $p_3 \xrightarrow{a_3} q_3$ and $p_4 \xrightarrow{a_4} q_4$. Moreover let q_1^u and q_2^u, q_1^v and q_2^v, q_1^w and q_2^w be the endpoints of the artificial transitions of u, v and w respectively.

Path P has to traverse one of the endpoints for all three lca's according to Proposition 1 (c). In particular, P may have the form

$$p_1 \xrightarrow{a_1} q_1 \xrightarrow{*} q_2^u \xrightarrow{*} p_2 \xrightarrow{a_2} q_2 \xrightarrow{*} q_1^v \xrightarrow{*} p_3 \xrightarrow{a_3} q_3 \xrightarrow{*} q_1^w \xrightarrow{*} p_4 \xrightarrow{a_4} q_4.$$

We concentrate on the two path fragments

$$q_2^u \xrightarrow{*} p_2 \xrightarrow{a_2} q_2 \xrightarrow{*} q_1^v \text{ and } q_1^v \xrightarrow{*} p_3 \xrightarrow{a_3} q_3 \xrightarrow{*} q_1^w.$$

If we introduce new ε-free transitions

$$q_2^u \xrightarrow{a_2} q_1^v \text{ and } q_1^v \xrightarrow{a_3} q_1^w,$$

then disregarding the very first and the very last ε-free transitions of P, we have utilized the lca's as shortcuts in the ε-free equivalent $q_2^u \xrightarrow{a_2} q_1^v \xrightarrow{a_3} q_1^w$ of P.

We now analyze this procedure in general. In particular we improve upon the conversion of [2], where $O(n \cdot \log_2 n \cdot k)$ transitions are shown to suffice. Our approach combines ideas in [2] with Proposition 1 (c).

Lemma 1. *Let R be a regular expression of length n over an alphabet of size k. Then there is an ε-free NFA N for R with $O(n \cdot \log_2 n \cdot \log_2 2k)$ transitions. N has a unique initial state. If $\varepsilon \notin L(R)$, then N has one final state and otherwise N has at most two final states.*

Proof. Assume that q_0 is the unique initial state of N_R and q_f its unique final state. We moreover assume without loss of generality that all states of N_R have an ε-loop. We choose q_0 and q_f as well as all endpoints of artificial transitions assigned to nodes of T_R^* as states for our ε-free NFA. q_0 is still the initial state and q_f is (jointly with q_0, whenever $\varepsilon \in L(R)$) the only final state. Thus the ε-free NFA results from N_R after removing all ε-transitions (and all states which are incident to ε-transitions only) and inserting new ε-free transitions.

Let $p \xrightarrow{a} q$ be an ε-free transition of N_R and let l be the corresponding leaf of T_R^*. We define the set A to contain q_0, q_f as well as all ancestors of l including l itself. We assume that q_0 and q_f are roots of imaginary trees such that *all* leaves of T_R^* belong to the "right subtree" of q_0 as well as to the "left subtree" of q_f.

Consider any two nodes $v, w \in A$ and let q_1^v, q_2^v and q_1^w, q_2^w be the endpoints of the artificial transitions for v and w respectively. (We set $q_1^v = q_2^v = q_0$ for $v = q_0$ and $q_1^v = q_2^v = q_f$ for $v = q_f$.) We insert the transition $\mathbf{q_i^v} \xrightarrow{\mathbf{a}} \mathbf{q_j^w}$ for $i, j \in \{1, 2\}$, if there are ε-paths $q_i^v \xrightarrow{*} p$ and $q \xrightarrow{*} q_j^w$ in N_R. Let N be the NFA obtained from N_R

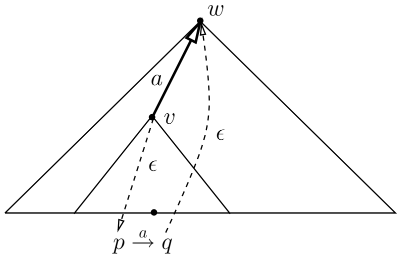


Fig. 4. Introducing transitions between ancestors

after these insertions and after removing all ε-transitions.

Obviously any accepting path from q_0 to q_f in N can be extended via ε-transitions to an accepting path in N_R and hence $L(N) \subseteq L(N_R)$. Now consider an accepting path

$$q_0 \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} p_1 \xrightarrow{a_1} q_1 \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} p_r \xrightarrow{a_r} q_r \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} q_f$$

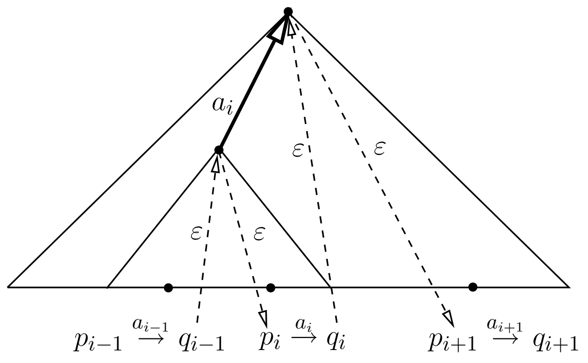
for the word $a_1 \circ \cdots \circ a_r$ in N_R. Since all states of N_R have ε-loops, we may assume that all ε-free transitions are separated by ε-paths. Let l_i be the leaf of T_R^* corresponding to the transition $p_i \xrightarrow{a_i} q_i$. To obtain an accepting path in N, let $v_0 = q_0, v_1, \ldots, v_{r-1}, v_r = q_f$ be a sequence of nodes, where v_i $(1 \leq i \leq r-1)$ is the lowest common ancestor of l_i and l_{i+1} in T_R^*. By Proposition 1(c) the ε-paths from q_{i-1} to p_i and from q_i to p_{i+1} have to hit endpoints $q_{j_{i-1}}^{v_{i-1}} \in \{q_1^{v_{i-1}}, q_2^{v_{i-1}}\}$ and $q_{j_i}^{v_i} \in \{q_1^{v_i}, q_2^{v_i}\}$ of the respective artificial transition. But then N contains the transition $q_{j_{i-1}}^{v_{i-1}} \xrightarrow{a_i} q_{j_i}^{v_i}$ and

$$q_0 \xrightarrow{a_1} q_{j_1}^{v_1} \xrightarrow{a_2} \cdots \xrightarrow{a_{i-1}} q_{j_{i-1}}^{v_{i-1}} \xrightarrow{a_i} q_{j_i}^{v_i} \xrightarrow{a_{i+1}} q_{j_{i+1}}^{v_{i+1}} \xrightarrow{a_{i+2}} \cdots \xrightarrow{a_r} q_f$$

is an accepting path in N. Hence $L(N_R) \subseteq L(N)$ and N and N_R are equivalent.

We still have to count the number of transitions of N. We introduce transitions $q_i^s \xrightarrow{a} q_j^t$ (resp. $q_i^t \xrightarrow{a} q_j^s$) only for transitions $p \xrightarrow{a} q$ which are represented by leaves belonging to the subtrees of s and t. Hence s must be an ancestor or a descendant of t in T_R^*. Thus, for given nodes s, t we introduce at most $\min\{|s|, |t|, k\}$ transitions, where $|s|$ and $|t|$ are the number of leaves in the subtrees of s and t respectively.

We fix s. There are $O(|s|/k)$ descendants t of s with $|t| \geq k$ and at most $O(\frac{|s|}{k} \cdot k) = O(s)$ newly introduced transitions connect s with a "high" node t. The remaining "low" nodes are partitioned into $O(\log_2 k)$ levels, where one level produces at most $O(|s|)$ transitions, since at most $|t|$ transitions connect s with a node t of the level. Thus the number of transitions between q_i^s and q_j^t for descendants t of s is bounded by $O(|s| \cdot (1 + \log_2 k))$ and hence by $O(|s| \cdot \log_2 2k)$. Finally we partition all nodes s of T_R^* into $O(\log_2 n)$ levels, where one level requires $O(n \cdot \log_2 2k)$ transitions, and overall $O(n \cdot \log_2 n \cdot \log_2 2k)$ transitions suffice. □



a_i
ε
ε
ε
ε
$p_{i-1} \xrightarrow{a_{i-1}} q_{i-1}$
$p_i \xrightarrow{a_i} q_i$
$p_{i+1} \xrightarrow{a_{i+1}} q_{i+1}$

Fig. 5. The construction of an equivalent path

2.3 A Recursive Construction of Small ε-Free NFAs

How can we come up with even smaller ε-free NFAs? Assume that we have partitioned the regular expression R into (very small) subexpressions of roughly same size η. We apply the construction of Lemma 1 to all subexpressions and introduce at most $O(n \cdot \log_2 \eta \cdot \log_2 2k)$ transitions, a significant reduction if η is drastically smaller than n. However now we have to connect different subexpressions with "global" transitions and Lemma 1 inserts the vast majority of transitions, leading to a total of $O(n \cdot \log_2 n \cdot \log_2 2k)$ transitions. But we can do far better, if we are willing to increase the size of ε-free NFAs for every subexpression S.

Definition 1. *Let N be an ε-free NFA with initial state q_0 and let F be the set of final states. We say that any transition (q_0, r) is an* ***initial transition*** *and that r is a* ***post-initial state****. Analogously any transition (r, s) is a* ***final transition****, provided $s \in F$, and r is a* ***pre-final state****.*

Observe that it suffices to connect a global transition for a subexpression S with a post-initial or pre-final state of an ε-free NFA for S. As a consequence, the number of global transitions for S is reduced drastically, provided we have only few post-initial and pre-final states. But, given an ε-free NFA, how large are equivalent ε-free NFAs with relatively few initial or final states?

Proposition 2. *Let N be an ε-free NFA with s transitions over an alphabet Σ of size k. Then there is an equivalent ε-free NFA N' with $O(k^2 + k \cdot s)$ transitions and at most $3k + k^2$ initial or final transitions. N' has one initial state. If $\varepsilon \notin L(N)$, then N' has one final state and otherwise at most two final states.*

Proof. Assume that q_0' is the initial state of N, F is the set of final states and $\Sigma = \{1, \ldots, k\}$. Let $\rho_1, \ldots, \rho_p$ be the post-initial states of N and $\sigma_1, \ldots, \sigma_q$ be the pre-final states of N. Moreover let R_i be the set of post-initial states in N receiving an i-transitions from q_0' and let S_i be the set of pre-final states of N sending an i-transitions into a state of F.

We introduce a new initial state q_0 and a new final state q_f. (q_0 is the second accepting state, if $\varepsilon \in L(N)$.) For every $a \in \Sigma \cap L(N)$ we insert the transition $q_0 \xrightarrow{a} q_f$. We introduce new post-initial states $r_1, \ldots, r_k$, new pre-final states $s_1, \ldots, s_k$ and insert i-transitions from q_0 to r_i as well as from s_i to q_f.

If $\rho_j \in R_i$ and if (ρ_j, s) is a transition with label b, then insert the transition (r_i, s) with label b. Analogously, if $\sigma_j \in S_i$ and if (r, σ_j) is a transition with label b, then we insert the transition (r, s_i) with label b. Thus the new states r_i and s_i inherit their outgoing respectively incoming transitions from the states they "are responsible for". Finally, to accept all words of length two in $L(N)$, we introduce at most k^2 further initial and final transitions incident with q_0, q_f and post-initial states.

Observe that the new NFA N' is equivalent with N, since, after leaving the new states r_i and before reaching the new states s_j, N' works like N. The states q_0 and q_f are incident with at most $k + 2 \cdot k + k^2$ transitions: up to k transitions link q_0 and q_f, $2 \cdot k$ transitions connect q_0 and the r_i's (or the s_i's and q_f) and at most k^2 transitions accept words of length two. Finally at most $k \cdot s$ transitions leave states r_i, not more than ks transitions enter states s_j and hence the number of transitions increases from s to at most $s \cdot (2k + 1) + 3 \cdot k + k^2$. □

We now observe that the combination of Lemma 1 and Proposition 2 provides a significant savings for alphabets of small size.

Proof of Theorem 1. Let T_R be the expression tree and T^*_R be the decomposition tree. If u is a node of T^*_R, then let $T^*_R(u)$ denote the subtree of T^*_R with root u. We begin with a sketch of the construction of a small ε-free NFA for the expression R.

We proceed iteratively. In a first phase we process a cut of "low" nodes of T^*_R, where any such node u has at most L_1 original transitions in its subtree $T^*_R(u)$, where L_1 will be fixed later. We insert additional transitions between descendants of u according to Lemma 1 and obtain an ε-free NFA $N^1(u)$. We repeat this procedure in phase j; this time the cut consists of nodes u with at most L_j original transitions in their respective subtree $T^*_R(u)$. The parameters L_j will be fixed later; here we only assume that $L_1 < \cdots < L_{j-1} < L_j < \cdots$ holds. Thus we process cuts of nodes of increasing height until we reach the root of T^*_R.

Let D be the set of descendants w of u which we processed in the previous phase. At the beginning of phase j we assume that all ε-free NFA $N^{j-1}(w)$ for $w \in D$ have been constructed. When constructing an ε-free NFA $N^j(u)$ we are now facing a more complicated situation than in Lemma 1. Firstly, when building $N^j(u)$ in a phase $j > 1$, we have to "merge" the ε-free NFA $N^{j-1}(w)$ for all $w \in D$. As opposed to the case of original transitions, any such NFA $N^{j-1}(w)$ may accept arbitrary strings instead of just single letters. To prepare for this more complicated scenario we first apply Proposition 2 to replace $N^{j-1}(w)$ by an equivalent ε-free NFA with few post-initial and pre-final states: all additional transitions can now be connected with one of these relatively few states.

Assume that u represents the expression tree $T^v_R(C)$. Then the ε-free NFA $N^j(u)$ will be built from $N^v_R(C \cup D)$. In particular, the artificial transitions corresponding to any $w \in D$ are replaced by the ε-free NFA $N^{j-1}(w)$. But, and this is the second difference to the situation of Lemma 1, the artificial transitions $q_1 \xrightarrow{S} q_2$ corresponding to a node in C are kept: this procedure allows to plug in the ε-free NFA N recognizing S, whenever N is constructed. In summary, $N^j(u)$ results from its "base NFA" $N^v_R(C \cup D)$ by removing all ε-free transitions, keeping all artificial transitions for nodes in C, replacing artificial transitions for nodes in D by the previously determined ε-free NFA and by adding new ε-free transitions. We have to deal with the last point in detail.

Our construction utilizes the following invariant: if $w \in D$ represents the expression tree $T^x_R(C')$, then the NFA $N^{j-1}(w)$ is equivalent with the NFA $N^x_R(C')$. Here we assume, for $N^{j-1}(w)$ as well as for $N^x_R(C')$, that any artificial transition $q_1 \xrightarrow{S} q_2$ "produces" all words of the regular expression S.

We do not differentiate between the first and subsequent phases, since subsequent phases are more general: in phase 1 only ε-free transitions are to be "merged". But we may interpret any such transition as a ε-free NFAs $N^0(w)$ of "phase 0" consisting of a single transition only. In particular, the invariant holds initially.

As in the proof of Lemma 1 we assume that all states of N_R have an ε-loop. We now begin the formal description of our construction.

2.3.1 Phase j. We consider all nodes u of T^*_R which have at most L_j original transitions in their subtree $T^*_R(u)$, whereas its parent has more than L_j original transitions. Obviously these nodes define a cut in T^*_R.

Let D be the set of descendants w of u which we processed in the previous phase. We build an ε-free NFA $N^j(u)$ from the ε-free NFAs $N^{j-1}(w)$ for all $w \in D$. For any such descendant w we apply Proposition 2 to $N^{j-1}(w)$ and obtain an ε-free NFA $N^{j-1}(w)^*$ with at most $O(k^2)$ initial or final transitions; moreover $\text{size}(N^{j-1}(w)^*) \leq O(k^2 + k \cdot \text{size}(N^{j-1}(w))$. We utilize the few initial or final transitions to cheaply interconnect $N^{j-1}(w)^*$ with (endpoints of artificial transitions assigned to) ancestors of w within $T^*_R(u)$.

Assume again that u represents the expression tree $T^v_R(C)$. Then the ε-free NFA $N^j(u)$ is obtained from $N^v_R(C \cup D)$ by removing all ε-transitions, replacing the artificial transitions assigned to a node $w \in D$ by the ε-free NFA $N^{j-1}(w)^*$, keeping all artificial transitions for nodes in C and adding additional ε-free transitions. Only the last point has to be explained. As in the construction of Lemma 1, $N^j(u)$ keeps the initial and final states q^u_1 and q^u_2 of $N^v_R(C \cup D)$.

The insertion of new transitions is now a more complex task than in the situation of Lemma 1, since we are working with full-fledged ε-free NFAs instead of ε-free transitions. In particular we have to differentiate three cases, namely firstly the new case $\varepsilon \in L(N^{j-1}(w)^*)$, then the original case considered in Lemma 1, namely $a \in L(N^{j-1}(w)^*)$ for some letter a, and finally the second new case, namely that $L(N^{j-1}(w)^*)$ contains words of length at least two.

Let q_1, q_2 be the unique initial and final states of $N^{j-1}(w)^*$.

(0) Assume $\varepsilon \in L(N^{j-1}(w)^*)$. This case establishes ε-paths and is of interest only for the two remaining cases where reachability by ε-paths is crucial.

(1) Assume $a \in L(N^{j-1}(w)^*)$ for some letter a. For any ancestors t_1, t_2 of w in $T^*_R(u)$, for any endpoints q^{t_1}, q^{t_2} of their respective artificial transitions and for any ε-paths $q^{t_1} \xrightarrow{*} q_1$ and $q_2 \xrightarrow{*} q^{t_2}$, introduce the transition $q^{t_1} \xrightarrow{a} q^{t_2}$. (This procedure is completely analogous to Lemma 1.)

(2) Let $q_1 \xrightarrow{a} r$ be an arbitrary initial transition of $N^{j-1}(w)^*$. Then, for any ancestor t of w in $T^*_R(u)$, for any endpoint q^t of its artificial transition and for any ε-path $q^t \xrightarrow{*} q_1$ introduce the transition $q^t \xrightarrow{a} r$. Analogously, if $r \xrightarrow{a} q_2$ is an arbitrary final transition of $N^{j-1}(w)^*$ and if there is an ε-path $q_2 \xrightarrow{*} q^t$, then introduce the transition $r \xrightarrow{a} q^t$.

We treat the artificial transitions of $T^v_R(C)$, respectively their ε-free NFA which will replace the artificial transitions at a later time, in exactly the same way.

We have to show that $N^j(u)$ satisfies the invariant, i.e., that $N^j(u)$ and $N^v_R(C)$ are equivalent whenever any artificial transition $q_1 \xrightarrow{S} q_2$ "produces" all words of the regular expression S. If we insert a transition $p \xrightarrow{a} q$, then there are states r, s and a path $p \xrightarrow{*} r \xrightarrow{a} s \xrightarrow{*} q$ in $N^v_R(C)$. Thus $L(N^j(u)) \subseteq L(N^v_R(C))$.

Any accepting path $\mathcal{P}$ in $N^v_R(C)$ traverses ε-free NFAs corresponding to some sequence $(w_1, \ldots, w_s)$ for $w_1, \ldots, w_s \in D$; we may require that at least one letter is read for each w_i. Since all states of N_R have ε-loops, we may assume that all ε-free transitions are separated by ε-paths. Hence, if w_i represents an NFA with initial state $q^{w_i}_1$ and final state $q^{w_i}_2$, then there is a ε-path from $q^{w_i}_2$ to $q^{w_{i+1}}_1$ in $N^v_R(C)$.

Assume that s and t are the least common ancestors in $T^*_R(u)$ of w_{i-1} and w_i and of w_i and w_{i+1} respectively. We apply Proposition 1 (c) and obtain ε-paths $q^s \xrightarrow{*} q^{w_i}_1$ and $q^{w_i}_2 \xrightarrow{*} q^t$ for appropriate endpoints q^s, q^t of the artificial transitions of s and t

respectively. If at least two letters are read for w_i then the corresponding accepting path $\mathcal{Q}$ in $N^j(u)$ jumps from q^s to a post-initial state of $N^{j-1}(w_i)$, then traverses $N^{j-1}(w_i)$ and finally jumps from a pre-final state of $N^{j-1}(w_i)$ to q^t; otherwise $\mathcal{Q}$ jumps from q^s directly to q^t. Thus $L(N_R^v(C)) \subseteq L(N^j(u))$: $N^j(u)$ and $N_R^v(C)$ are equivalent.

2.3.2 Accounting.

In phase j we are interconnecting not only the ε-free automata $N^{j-1}(w)$ of the previous phase, but also (the NFAs constructed for) the artificial transitions. We therefore begin our analysis by comparing the number of artificial transitions used for phase j with the size of the cut for phase j.

Remember that a node u belongs to the cut for phase j iff u has at most L_j original transitions in its subtree $T_R^*(u)$, but its parent p has more than L_j original transitions in its subtree $T_R^*(p)$.

Proposition 3. *(a) The ancestors of cut nodes define a binary tree C_j with cut nodes as the set of leaves.*

(b) The number of different artificial transitions generated at a proper ancestor of a cut node is smaller than the size of the cut.

(c) The total number of artificial transitions used when constructing all ε-free NFA $N^j(u)$ of phase j is not larger than the size of the cut processed in phase $j-1$.

Proof. (a) Let v be an ancestor of a cut node. If v does not belong to the cut, then v has more than L_j original transitions and hence v has two children which are ancestors of cut nodes. (Here we assume that a node is its own ancestor.)

(b) Since an ancestor generates exactly one artificial transition, there are no more artificial transitions than there are inner nodes of C_j. According to part (a), C_j is a binary tree and hence the number of inner nodes is smaller than the number of leaves, i.e., smaller than the size of the cut.

(c) Observe first that an artificial transition occurs in at most one cut node: if an artificial transition is a descendant of the separating node, it appears only in the left subtree and otherwise only in the right subtree. The claim follows now from part (b). □

Thus we may only count the number of ε-free transitions introduced because of the ε-free NFA $N^{j-1}(w)$ and may disregard artificial transitions all together.

We have partitioned the new ε-free transitions in two classes. To count transitions from class (1), observe that $T_R^*(u)$ has at most $O(\log_2 L_j)$ levels and the NFA $N^j(u)$ has to merge at most $O(\frac{L_j}{L_{j-1}})$ descendant NFAs $N^{j-1}(w)$. We may now apply the analysis developed for Lemma 1 to receive the upper bound $O(\frac{L_j}{L_{j-1}} \cdot \log_2 L_j \cdot \log_2 2k)$ on the number of class 1 transitions introduced for $N^j(u)$.

The transitions in class (2) connect one of the $O(k^2)$ post-initial and pre-final states of some N_w^* with endpoints of artificial transitions for at most $O(\log_2 L_j)$ ancestors within $T_R^*(u)$. Thus for each NFA $N^j(u)$ we have introduced at most $O(k^2 \cdot \frac{L_j}{L_{j-1}} \cdot \log_2 L_j)$ transitions from the second class. If each descendant NFA $N^{j-1}(w)$ has size at most s_{j-1}, then all descendants contribute not more than $O(k \cdot \frac{L_j}{L_{j-1}} \cdot s_{j-1})$ transitions including the blow-up due to Proposition 2. Hence $N^j(u)$ has at most $O(s_j)$ transitions, where

$$\begin{aligned} s_j &= k \cdot \frac{L_j}{L_{j-1}} \cdot s_{j-1} + \frac{L_j}{L_{j-1}} \cdot \log_2 L_j \cdot \log_2 2k + k^2 \cdot \frac{L_j}{L_{j-1}} \cdot \log_2 L_j \\ &\leq k \cdot \frac{L_j}{L_{j-1}} \cdot s_{j-1} + 2k^2 \cdot \frac{L_j}{L_{j-1}} \cdot \log_2 L_j. \qquad (1) \end{aligned}$$

We iterate recurrence (1) and get for $1 \leq r \leq j$

$$s_j \leq k^r \cdot \frac{L_j}{L_{j-r}} \cdot s_{j-r} + \sum_{t=0}^{r-1} 2k^{t+2} \cdot \frac{L_j}{L_{j-1-t}} \cdot \log_2 L_{j-t}.$$

Assume that the regular expression R has length n. Thus, if we assume that $n = L_i$ and set $r = j = i$, then we introduce at most $O(s_i)$ transitions, where

$$s_i \leq k^i \cdot \frac{n}{L_0} \cdot s_0 + \sum_{t=0}^{i-1} 2k^{t+2} \cdot \frac{n}{L_{i-t-1}} \cdot \log_2 L_{i-t}. \tag{2}$$

Since phase 1 starts from single "phase 0" NFAs consisting of a single transition, we may set $L_0 = s_0 = 1$ and the first term of (2) coincides with $O(k^i \cdot n)$. We set $L_j = 2^{L_{j-1}}$ and the sum in (2) is bounded by $O(k^{i+1} \cdot n)$. Thus $i \leq \lceil \log^* n \rceil$ and the regular expression R is recognized by an ε-free NFA with at most $s_i = O(k^{1+\log^* n} \cdot n)$ transitions. □

3 The Lower Bound

We consider the regular expression

$$E_n = (1 + \varepsilon) \circ (2 + \varepsilon) \circ \cdots \circ (n + \varepsilon)$$

of strictly increasing sequences. The following lower bound is asymptotically optimal and improves upon the $\Omega(n \cdot \log_2^2 n / \log_2 \log_2 n)$ bound of [8].

Lemma 2. *ε-free NFAs for E_n have at least $\Omega(n \cdot \log_2^2 n)$ transitions.*

Before giving a proof we show that Theorem 2 is a consequence of Lemma 2. We concatenate E_k exactly n/k times with itself to obtain $R_{n,k} = (E_k)^{n/k}$.

Now assume that $N_{n,k}$ is an ε-free NFA recognizing $R_{n,k}$. We say that a transition e of $N_{n,k}$ *belongs to copy i* iff e is traversed by an accepting path with label sequence $(1 \circ 2 \circ \cdots \circ k)^{i-1} \circ \sigma \circ (1 \circ 2 \circ \cdots \circ k)^{n/k-i}$ while reading the string $\sigma \neq \varepsilon$. Now assume that there is a transition e which belongs to two different copies i, j with $i < j$. Then we can construct an accepting path with label sequence $(1 \circ 2 \circ \cdots \circ k)^{j-1} \circ \tau \circ (1 \circ 2 \circ \cdots \circ k)^{n/k-i}$ and $N_{n,k}$ accepts a word outside of $R_{n,k}$.

Thus any transition belongs to at most one copy. $N_{n,k}$ has, as a consequence of Lemma 2, at least $\Omega(k \cdot \log_2^2 k)$ transitions for each copy and hence $N_{n,k}$ has at least $\Omega(\frac{n}{k} \cdot k \cdot \log_2^2 k)$ transitions. Observe that the unary regular expression 1^n requires NFAs of linear size and hence we actually get the lower bound $\Omega(\frac{n}{k} \cdot k \cdot \log_2^2 2k)$ also for $k = 1$.

3.1 An outline of the argument

Let $n = 2^k$. and let N_n be an arbitrary ε-free NFA for E_n. Our basic approach follows the argument of [8]. In particular, we may assume that N_n is in *normal form*, i.e., $\{0, 1, \ldots, n\}$ is the set of states of N_n and any transition $i \xrightarrow{l} j$ satisfies $i < l \leq j$.

To study the behavior of transitions we introduce the ordered complete binary tree T_n with nodes $\{1, \ldots, n-1\}$ and depth $k-1$. We assign names to nodes such that an inorder traversal of T_n produces the sequence $(1, \ldots, n-1)$. Finally we label the root r of T_n with the set $L(r) = \{1, \ldots, n\}$. If node v is labeled with the set $L(v) = \{i+1, \ldots, i+2t\}$, then we label its left child v_l with $L(v_l) = \{i+1, \ldots, i+t\}$ and its right child v_r with $L(v_r) = \{i+t+1, \ldots, i+2t\}$. Finally define $|v| = |L(v)|$ as the size of v. Observe that $|v| = 2$ holds for every leaf v and we interpret its children v_l, resp. v_r as "virtual leaves".

Example 1. Again set $n = 2^k$. We recursively construct a family of ε-free NFAs A_n to recognize E_n. $\{0, 1, \ldots, n-1, n\}$ is the set of states of A_n; state 0 is the initial and state n is the final state of A_n. To obtain A_n place two copies of $A_{n/2}$ in sequence: $\{0, 1, \ldots, n/2 - 1, n/2\}$ and $\{n/2, n/2 + 1, \ldots n - 1, n\}$ are the sets of states of the first and second copy respectively, where the final state $n/2$ of the first copy is also the initial state of the second copy.

If $(a_1, \ldots, a_r, a_{r+1}, \ldots a_s)$ is any increasing sequence with $a_r \leq n/2 < a_{r+1}$, then the sequence has an accepting path which starts in 0, reaches state $n/2$ when reading a_r and ends in state n when reading a_s. But increasing sequences ending in a letter $a \leq n/2$, resp. starting in a letter $a > n/2$ have to be accepted as well. Therefore direct all transitions, ending in the final state $n/2$ of the first copy, also into the final state n. Analogously, direct all transitions, starting from the initial state $n/2$ of the second copy, also out of initial state 0.

Now unroll the recursion and visualize A_n on the tree T_n (after disregarding the initial state 0 and the final state n). The root of T_n plays the role of state $n/2$. In particular, for any node v there are $|v|$ many transitions with labels from the set $L(v)$ between v and the root. Thus the root is the target of $n \cdot k = n \cdot \log_2 n$ transitions, implying that A_n has $n \cdot \log_2^2 n$ transitions if transitions incident with states 0 or n are disregarded.

Definition 2. *We say that node v of T_n is crossed from the left in N_n iff for all $i \in L(v_l)$ and all sequences σ with $\sigma \circ i \in E_n$ there is a path in N_n with label sequence $\sigma \circ i$ which ends in a state $y \in L(v_r)$.*

If all sequences $i \circ \tau \in E_n$ with arbitrary $i \in L(v_r)$ have a path which starts in some state $x \in L(v_l)$, then we say that v is crossed from the right.

In particular the last transition of the path "crosses" v, since it ends in $L(v_r)$ and is labeled with a letter from $L(v_l)$.

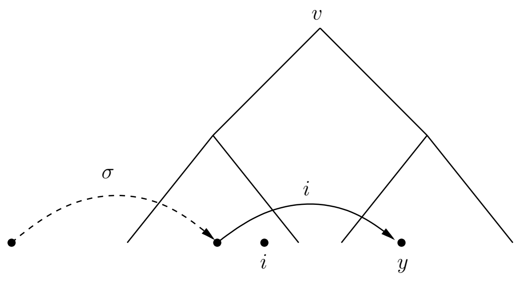


Fig. 6. An i-transition crossing v from the left

Proposition 4. *[8] Let v be an arbitrary node of T_n. Then for any ε-free NFA in normal form, v is crossed from the left or v is crossed from the right.*

Proof. Assume that v is not crossed from the left. Then there is a word $\sigma \circ i \in E_n$ with $i \in L(v_l)$ such that no path in N_n with label sequence $\sigma \circ i$ has a final transition crossing v. If v is also not crossed from the right, then there is a word $j \circ \tau \in E_n$ with $j \in L(v_r)$ such that no path in N_n with label sequence $j \circ \tau$ has an initial transition crossing v. But then N_n rejects $\sigma \circ i \circ j \circ \tau \in E_n$. □

Let C be the set of nodes $v \in T_n$ which are crossed from the left. We assume that "more" nodes are crossed from the left and hence we concentrate on C. (See (6) in Section 3.3 for a formal definition of "more").

Assume that $w \in T_n$ belongs to C and that node v belongs to Left(w), the set of nodes of T_n which belong to the left subtree of w. Then any sequence $\sigma \circ j \in E_n$ with $j \in L(v_r)$, and hence $j \in L(w_l)$, has a path $p_{\sigma,j}$ in N_n with label sequence $\sigma \circ j$ which ends in a state $y \in L(w_r)$. Observe that the last transition $e = (x, y)$ of $p_{\sigma,j}$ identifies w as the unique tree node with $j \in L(w_l)$ and $y \in L(w_r)$. Moreover, if $x \in L(v_l)$, then e also identifies v as the unique tree node with $x \in L(v_l)$ and $j \in L(v_r)$.

We now observe that N_n has $\Omega(n \cdot \log_2^2 n)$ transitions if a majority of pairs (v, w) with $v \in$ Left(w) is identified for too many labels $j \in L(v_r)$. In particular, define $N(h, h')$ for $h' < h \leq k - 1$ as the number of pairs (j, w), where $w \in T_n$ has height h and j belongs to the right subtree of a node $v \in$ Left(w) with height h'. Then $N(h, h') = n/4$, since for any w exactly one fourth of all labels j is counted. But then $\sum_{h'<h\leq k-1} N(h, h') = \Omega(n \cdot \log_2^2 n)$ holds and it suffices to show that each pair (v, w) with $w \in C$ and $v \in$ Left(w) has $\Omega(|v_r|)$ transitions which identify v as well as w.

Labels $j \in L(v_r)$ are problematic if all j-transitions $e = (x, y)$ with $x \in L(v)$ and $y \in L(w_r)$ are *short* for (v, w), i.e., any such transition e starts in $x \in L(v_r)$. If label $j \in L(v_r)$ is short, then j-transitions "into" $L(w_r)$ depart close to j.

> But, since w is crossed from the left, preceding i-transition, for $i \in L(u)$ with $u \in$ Left(v), have to reach one of these starting points, and if these starting points are "close to home" for many short labels j, then consequently many copies of i-transitions are required.

To formalize this intuition we determine how far to the left short j-transitions extend, but not with respect to transitions starting in $L(v)$ and ending in $L(w_r)$ for some specific w, but rather with respect to a worst-case sequence $\tau = j \circ \sigma \circ k \in E_n$ (with $k \in L(w_l)$ for an arbitrary $w \in C$) such that any path with sequence τ starts very close to j, if we require the path to start in $L(v)$ and to end in $L(w_r)$.

Definition 3. *Vertices $v \in T_n$, $w \in C$ (with $v < w$) as well as labels $j \in L(v_r)$ and $k \in L(w_l)$ are given. Define*

$$\mathrm{d}_{v,w}(j, k) = \min_{\tau = j\circ\sigma\circ k \in E_n} \; \max_{x \in L(v), y \in L(w_r)} \{\, j - x \mid \exists \textit{ path } x \xrightarrow{*} y \textit{ with sequence } \tau \}$$

and $\mathrm{d}_v(j) = \min_{w \in C, k \in L(w_l)} d_{v,w}(j, k)$.

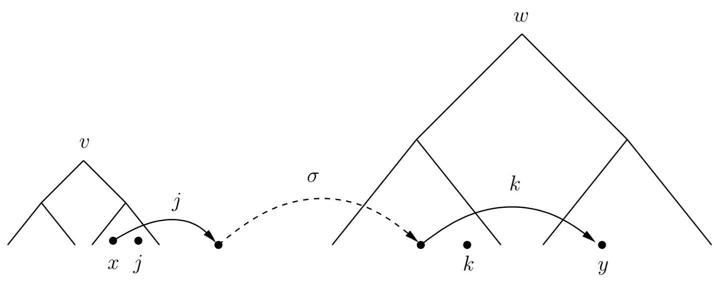


Fig. 7. Measuring the minimal distance of starting points x of j-transitions from j

Observe first that $\mathrm{d}_v(j)$ is only defined iff there is a node $w \in C$ with $v < w$. But we can assume that the node $w = n - 1$ is crossed from the left: $L(w_l) = \{n - 1\}$ holds and any transition with label $n - 1$ has to either end in node w or in the virtual leaf w_r. Thus if we copy all transitions with label $n-1$ "from w to w_r", then w is crossed from the left at the cost of at most doubling the size of the NFA.

For any $i \in L(u)$ with $u \in \text{Left}(v)$ there has to be an i-transition which approaches j within distance at most $\mathrm{d}_v(j)$. Next we determine how close the majority of labels $j \in L(v_r)$ have to be approached.

Definition 4. *Let s be maximal with the property that at least $\frac{|v_r|}{2}$ labels $j \in L(v_r)$ satisfy $\mathrm{d}_v(j) \leq \frac{|v_r|}{s}$. Set $s(v) = s$ and call a label $j \in L(v_r)$ regular for v iff $\mathrm{d}_v(j) \leq \frac{|v_r|}{s(v)}$ holds.*

If $j \in L(v_r)$, then $\mathrm{d}_{v,w}(j,k) \leq 2 \cdot |v_r|$, since the starting point x of a j-transition belongs to $L(v)$. But then $d_v(j) \leq 2 \cdot |v_r|$ and

$$s(v) \geq 1/2 \tag{3}$$

follows, since $d_v(j) \leq 2 \cdot |v_r|$ for all labels $j \in L(v_r)$.

At least one half of all labels $j \in L(v_r)$ are regular, i.e., have j-transitions which are forced by some node $w \in C$ to have starting points within distance at most $\frac{|v_r|}{s(v)}$ from j. Now, if v is crossed from the left and if u belongs to $\text{Left}(v)$, then at least $\Omega(s(v))$ i-transitions end in v_r for all labels $i \in L(u)$: all regular labels $j \in L(v_r)$ have to be approached within distance at most $|v_r|/s(v)$.

All in all $|u| \cdot s(v)$ transitions are required for fixed u and v. Any such transition identifies v, however the same transition may be counted for several nodes $u \in \text{Left}(v)$. In particular we show

Lemma 3. *N_n has at least $\Omega(\sum_{v \in C} \sum_{u \in \text{Left(v)}} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))})$ transitions.*

We prove Lemma 3 in the next section and show in Section 3.3 that Lemma 2 is a consequence of Lemma 3.

3.2 Short Transitions

Let u be an arbitrary node. Then less than $\frac{|u_r|}{2}$ labels $i \in L(u_r)$ satisfy $\mathrm{d}_u(i) \leq \frac{|u_r|}{2s(u)}$ and hence

$$\frac{|u_r|}{2s(u)} < d_u(i) \tag{4}$$

holds for more than one half of all labels $i \in L(u_r)$.

Proof of Lemma 3. We arbitrarily pick nodes $u \in T_n, v \in C$ with $u \in \text{Left}(v)$ and a regular label j for v. Then there is a node $w \in C$ with $v < w$ and a label sequence $\tau^* = j \circ \tau \circ k \in E_n$ with $k \in L(w_l)$ such that each path for τ^*, which begins in $x \in L(v)$ and ends in $L(w_r)$, satisfies $j - x \leq |v_r|/s(v)$.

Let h be the smallest label in $L(u_l)$. If i belongs to $L(u_r)$, then any path with label sequence $h \circ i \circ \tau^*$ which ends in $L(w_r)$, has to have an i-transition e which starts in $L(u)$ (since $h \in L(u_l)$) and ends in $L(v)$ (since $u \in \text{Left}(v)$ and $j \in L(v)$). From all i-transitions which belong to a path $u \xrightarrow{*} w_r$ with label sequence $i \circ \tau^*$, we select an i-transition $e = (x, y)$ with smallest possible left endpoint $x \in L(u)$ and call e *distinguished* (for (u, v)).

When counting transitions of N_n we restrict ourselves to distinguished transitions. Firstly we determine the number of distinguished i-transitions for $i \in L(u_r)$ which start in $L(u)$ and end in $L(v_r)$. Secondly we bound the effect of multiple counting: all i-transition do start in $L(u) \subseteq L(v_l)$ and hence they identify v, whenever they end in $L(v_r)$. However i-transitions may not identify u.

At most $|v_r|/s(v)$ regular labels j for v have j-transitions with a common left endpoint. Moreover at most $|v_r|/s(v)$ regular labels have transitions with left endpoint in $L(v_l)$ and therefore at least

$$\frac{|v_r|/2 - |v_r|/s(v)}{|v_r|/s(v)} = \frac{|v_r|/2}{|v_r|/s(v)} - 1 \geq \frac{s(v)}{2} - 1$$

different left endpoints in $L(v_r)$ are required. As a consequence, for all $i \in L(u_r)$, at least $s(v)/2 - 1$ distinguished i-transitions start in $L(u)$ and end in $L(v_r)$. This result is meaningless if $s(v) < 2$, but since $v \in C$, N_n has for every label $i \in L(u_r)$ a path with label sequence $h \circ i$, where the i-transition starts in $L(u)$ and ends in $L(v_r)$. Thus

for all $i \in L(u_r)$ at least $\max\{s(v)/2 - 1, 1\} \geq s(v)/4$ distinguished i-transitions start in $L(u)$ and end in $L(v_r)$. All these transitions identify v by their left and right endpoint.

Let $E(u,v)$ be the set of transitions of N_n which are distinguished for u and v. We have just seen that $|E(u,v)| \geq |u_r| \cdot s(v)/4$ holds. Transitions in $E(u,v)$ identify v, however they may not identify u. In particular, for $i \in L(u_r)$ let $e = (x,y)$ be a distinguished i-transition for u and v. If $v \in C$, then

$$d_u(i) \leq d_{u,v}(i,j) \leq i - x, \tag{5}$$

holds, since distinguished transitions maximize the difference between their label i and their left endpoint (among all i-transitions participating in a path $u \overset{*}{\rightarrow} w_r$ which ends in a j-transition). Furthermore let μ be a left descendant of v of smallest depth such that the distinguished i-transition e is also distinguished for μ and v. Observe that u has to be a descendant of μ and hence e is distinguished for at most $\lceil \log_2 \frac{|\mu|}{i-x} \rceil$ nodes u. Thus, in order to control multiple counting, we have to bound $i - x$ from below. We apply (4) and (5) and obtain

$$\frac{|\mu_r|}{2 \cdot s(\mu)} < d_\mu(i) \leq i - x$$

for at least one half of all labels $i \in L(\mu_r)$. Therefore transition e belongs to at most $\lceil \log_2 \frac{|\mu|}{i-x} \rceil \leq \lceil \log_2 |\mu| / \frac{|\mu_r|}{2 \cdot s(\mu)} \rceil = \lceil \log_2 (4s(\mu)) \rceil$ sets $E(u,v)$. To avoid multiple counting we assign weight $1/\lceil \log_2(4s(u)) \rceil^2$ to transition $e \in E(u,v)$. If $\mu_1, \ldots, \mu_{r-1}, \mu_r = \mu$ are all the tree nodes in Left(v) for which e is distinguished and if μ_i is a descendant of μ_{i+1}, then $i \leq \lceil \log_2(4s(\mu_i)) \rceil$ and hence $\sum_{i=1}^{r} 1/\lceil \log_2(4s(\mu_i)) \rceil^2 \leq \sum_{i=1}^{r} 1/i^2 = O(1)$.

To summarize: we have $|E(u,v)| \geq |u_r| \cdot s(v)/4$ and there is no multiple counting, if we assign the weight $1/\log_2^2(4s(u))$ to transitions in $E(u,v)$. Hence N_n has asymptotically at least

$$\sum_{v \in C} \sum_{u \in \text{Left}(v)} \frac{|E(u,v)|}{\log_2^2(4s(u))} = \Omega\Big(\sum_{v \in C} \sum_{u \in \text{Left(v)}} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))}\Big)$$

transitions and the claim follows. □

3.3 Accounting

We have assumed in the proof sketch that "more" nodes are crossed from the left. We now formalize this to mean

$$\sum_{v \in C, \text{depth}(v) \geq (\log_2 n)/2} \frac{|v|}{2} \cdot \log_2 |v| \geq \sum_{v \notin C, \text{depth}(v) \geq (\log_2 n)/2} \frac{|v|}{2} \cdot \log_2 |v|. \tag{6}$$

If (6) does not hold, then we work instead with $\overline{C}$, the set of nodes which are crossed from the right. Thus we may assume (6). We set $\mathcal{T} = \sum_{\mathrm{depth}(v) \geq (\log_2 n)/2} \frac{|v|}{2} \cdot \log_2 |v|$ and observe that

$$\mathcal{T} = \sum_{d=(\log_2 n)/2}^{\log_2 n} \sum_{\mathrm{depth}(v)=d} \frac{|v|}{2} \cdot \log_2 |v| = \sum_{d=(\log_2 n)/2}^{\log_2 n} \frac{n}{2} \cdot d = \Omega(n \cdot \log_2^2 n)$$

holds. Thus Lemma 2 is a consequence of Lemma 3, if we show

$$\sum_{v \in C} \sum_{u \in \mathrm{Left(v)}} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))} = \Omega(\mathcal{T}), \tag{7}$$

which in turn follows, if

$$\sum_{u \in \mathrm{Left(v)}} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))} = \Omega(|v| \log_2 |v|) \tag{8}$$

holds for *sufficiently many nodes v*. We first formalize what "sufficiently many nodes" means.

Definition 5. *(a) $w(u) = \frac{|u|}{2} \cdot \log_2 |u|$ is the weight of u and*

$$q_v(u) = \frac{w(u)}{\sum_{u \in \mathrm{Left}(v)} w(u)}$$

is the probability of u with respect to v, provided u belongs to $\mathrm{Left}(v)$. *We define the probability $prob_v[E]$ of event $E \subseteq \mathrm{Left}(v)$ by the distribution q_v.*
(b) K is a suitably large constant with $\frac{1}{4} \cdot 2^{\sqrt{16k/\log_2 k}} \geq 2^{k^{1/3}}$ for all $k \geq K$. Finally set $p(v) = 16/\log_2 K$, if $s(v) \leq K$, and $p(v) = 16/\log_2 s(v)$ otherwise.

We begin by verifying (8) for all nodes v which qualify: we *disqualify* v iff $\mathrm{depth}(v) < (\log_2 n)/2$ or $v \notin C$ or if

$$\mathrm{prob}_v[s(u) \leq \frac{1}{4} \cdot 2^{\sqrt{p(v) \cdot s^*(v)}} \mid u \in \mathrm{Left}(v)] < p(v), \tag{9}$$

where $s^*(v) = \max\{s(v), K\}$. If we disqualify v for the last reason, then we also disqualify all descendants $u \in \mathrm{Left}(v)$ with $s(u) \leq 2^{\sqrt{p(v) \cdot s^*(v)}}/4$. Thus v is disqualified either for obvious reasons (i.e., $\mathrm{depth}(v) < (\log_2 n)/2$ or $v \notin C$) or if too few left descendants u have sufficiently small s^*-values and hence if (8) "seems" to be false for v.

In a second step we have to show that a node v is disqualified with sufficiently small probability. This is not surprising, since, if v is disqualified for non-obvious reasons, then $s(u)$ is extremely large in comparison to $s(v)$ for an overwhelming majority of left descendants u of v. To later disqualify a left "majority-descendant" u is now even harder, since $p(u)$ is inversely proportional to $\log_2 s(u)$.

We begin by investigating nodes which qualify.

Lemma 4. *Assume that v qualifies. Then*

$$\sum_{u \in \mathrm{Left}(v)} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))} \geq \frac{w(v)}{8K}.$$

Proof. Since v qualifies, we know that v belongs to C and $\mathrm{depth}(v) \geq (\log_2 n)/2$ holds. Moreover $s(u) \leq 2^{\sqrt{p(v)\cdot s^*(v)}}/4$ holds with probability $q \geq p(v)$ and hence $\log_2^2(4s(u)) \leq p(v) \cdot s^*(v)$ holds with probability q. We set

$$p_d(v) = \mathrm{prob}_v[s(u) \leq 2^{\sqrt{p(v)\cdot s^*(v)}}/4 \mid u \in \mathrm{Left}(v),\ \mathrm{depth}(u) = d] \ \text{ and}$$
$$p_d = \mathrm{prob}_v[\ \mathrm{depth}(u) = d \mid u \in \mathrm{Left}(v)].$$

Then $q = \sum_d p_d(v) \cdot p_d \geq p(v)$. We obtain

$$\sum_{u\in \mathrm{Left}(v)} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))} = \sum_{d=0}^{\log_2|v|-1} \sum_{u\in\mathrm{Left}(v),\ \mathrm{depth}(u)=d} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))}$$
$$\geq \sum_{d=0}^{\log_2|v|-1} p_d(v) \sum_{u\in\mathrm{Left}(v),\ \mathrm{depth}(u)=d} |u| \cdot \frac{s(v)}{p(v)\cdot s^*(v)}, \quad (10)$$

since $1/\log_2^2(4s(u)) \geq 1/(p(v)\cdot s^*(v))$ holds with probability at least $p_d(v)$ for nodes $u \in \mathrm{Left}(v)$ with $\mathrm{depth}(u) = d$. Thus we can further simplify the right hand side of (10) and get

$$\sum_{u\in\mathrm{Left}(v)} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))} \geq \frac{s(v)}{s^*(v)} \cdot \frac{|v|}{2} \cdot \sum_{d=0}^{\log_2|v|-1} \frac{p_d(v)}{p(v)}.$$

But

$$p_d = \frac{|v|/2 \cdot d}{\sum_{i=0}^{\log_2|v|-1} |v|/2 \cdot i} = \frac{d}{\sum_{i=0}^{\log_2|v|-1} i} \leq \frac{2\log_2|v|}{(\log_2|v|-1)\cdot\log_2|v|} \leq \frac{4}{\log_2|v|}$$

and hence $\sum_d 4 \cdot p_d(v)/\log_2|v| \geq \sum_d p_d(v) \cdot p_d \geq p(v)$, resp. $\sum_d p_d(v)/p(v) \geq (\log_2|v|)/4$. Thus we obtain

$$\sum_{u\in\mathrm{Left}(v)} |u| \cdot \frac{s(v)}{\log_2^2(4s(u))} \geq \frac{s(v)}{s^*(v)} \cdot \frac{|v|}{2} \cdot \sum_{d=0}^{\log_2|v|-1} \frac{p_d(v)}{p(v)} \geq \frac{s(v)}{s^*(v)} \cdot \frac{w(v)}{4}.$$

If $s(v) \geq K$, then $s(v) = s^*(v)$ and we gain the contribution $w(v)/4$. Otherwise $s(v) \leq K$ and we obtain at least the contribution $\frac{1/2}{K} \cdot \frac{w(v)}{4} = \frac{w(v)}{8K}$, since $s(v) \geq 1/2$ according to (3). Thus we have reached our goal of a contribution of at least $\frac{w(v)}{8K}$ in both cases. □

It suffices to show that sufficiently many nodes v qualify. If v is disqualified because of (9), then we lose the contribution $w(v) + p(v)\sum_{u\in\mathrm{Left}(v)} w(u)$, since we not only loose v, but possibly also a $p(v)$-fraction of all left descendants of v. How large is this loss?

Proposition 5. *For any node v,*

$$w(v) + p(v) \sum_{u\in\mathrm{Left}(v)} w(u) \leq 2p(v) \sum_{u\in\mathrm{Left}(v)} w(u). \quad (11)$$

Proof. We first observe

$$\sum_{u\in\mathrm{Left}(v)} w(u) = \sum_{d=0}^{\log_2|v|-1} \sum_{u\in\mathrm{Left}(v),\ \mathrm{depth}(u)=d} \frac{|u|}{2} \cdot d$$
$$= \sum_{d=0}^{\log_2|v|-1} \frac{|v|}{4} \cdot d = \frac{|v| \cdot \log_2|v| \cdot (\log_2|v|-1)}{8} \geq w(v) \cdot \frac{\log_2|v|}{8}$$

and therefore $w(v) \leq 8 \sum_{u \in \mathrm{Left}(v)} w(u) / \log_2 |v|$ follows. Hence the contribution we lose in a disqualification step for node v is bounded by

$$\begin{aligned} w(v) + p(v) \sum_{u \in \mathrm{Left}(v)} w(u) &\leq \left(\frac{8}{\log_2 |v|} + p(v)\right) \sum_{u \in \mathrm{Left}(v)} w(u) \\ &\leq 2p(v) \sum_{u \in \mathrm{Left}(v)} w(u), \end{aligned}$$

since $8/\log_2 |v| \leq 16/\log_2 n \leq p(v)$. □

We are now ready to bound the probability of disqualifying nodes.

Lemma 5. *A node v with depth$(v) \geq (\log_2 n)/2$ is disqualified with probability at most $\frac{1}{2} + \frac{64}{\log_2 K}$.*

Proof. Due to (6), a node v (with depth$(v) \geq (\log_2 n)/2$) is disqualified based on non-membership in C with probability at most $1/2$. Thus it suffices to show that disqualification because of (9) occurs with probability at most $64/\log_2 K$.

We order the disqualification steps for the nodes w according to increasing depth of w. (If a node is disqualified as a consequence of an earlier disqualification, then this node is not listed.) Now assume that node v is disqualified followed by a later disqualification of a left descendant $\overline{u}$ of v. Remember that with v we also disqualify all left descendants u with $s(u) \leq \frac{1}{4} \cdot 2^{\sqrt{p(v) \cdot s^*(v)}}$. But $\frac{1}{4} \cdot 2^{\sqrt{p(v) \cdot s^*(v)}} = \frac{1}{4} \cdot 2^{\sqrt{16 s^*(v)/\log_2 s^*(v)}}$ and by the choice of K, $\frac{1}{4} \cdot 2^{\sqrt{16k/\log_2 k}} \geq 2^{k^{1/3}}$ for all $k \geq K$. As a consequence, if the left descendant $\overline{u}$ of v has survived the disqualification step of v, then $s(\overline{u}) > 2^{s^*(v)^{1/3}}$ and in particular $p(\overline{u}) \leq p(v)/2$ follows.

If node v is disqualified, then we lose the contribution $2p(v) \cdot \sum_{u \in \mathrm{Left}(v)} w(u)$ due to (11). But then all later disqualification steps for left descendants of v result in a combined contribution of at most $(\sum_{i \geq 0} 2p(v)/2^i) \cdot \sum_{u \in \mathrm{Left}(v)} w(u) \leq 4p(v) \cdot \sum_{u \in \mathrm{Left}(v)} w(u)$.

We are considering a sequence of disqualified nodes, where the nodes are ordered according to increasing depth. If we double the "loss contribution" of a node v, i.e., if we assume the loss $4p(v) \cdot \sum_{u \in \mathrm{Left}(v)} w(u)$ for node v, then we may demand that no node in the sequence is a left descendant of another node in the sequence. The loss measured according to (11) is then maximized, if we disqualify all nodes of the rightmost path starting in the root r and if we assume that $s(v) \leq K$ for all nodes of the path. Obviously the overall loss is then bounded by

$$(4p(r) \cdot \sum_{i \geq 0} 2^{-i}) \cdot \sum_{u \in \mathrm{Left}(r)} w(u) \leq 4 \cdot \frac{16}{\log_2 K} \cdot 2 \sum_{u \in \mathrm{Left}(r)} w(u).$$

In other words, a node is disqualified with probability at most $\frac{4 \cdot 16}{\log_2 K}$. □

Lemma 2 is now an immediate consequence of Lemma 4 and Lemma 5, if K is chosen sufficiently large.

4 Conclusions and Open Problems

We have shown that every regular expression R of length n over an alphabet of size k can be recognized by an ε-free NFA with $O(n \cdot \min\{\log_2 n \cdot \log_2 2k, k^{1+\log^* n}\})$ transitions. For alphabets of fixed size (i.e., $k = O(1)$) our result implies that $O(n \cdot 2^{O(\log_2^* n)})$ transitions and hence almost linear size suffice. We have also shown the lower bound $\Omega(n \cdot \log_2^2 2k)$ and hence the construction of [7] is optimal for large alphabets, i.e., if $k = n^{\Omega(1)}$.

A first important open question concerns the binary alphabet. Do ε-free NFAs of linear size exist or is it possible to show a super-linear size lower bound? Moreover, although we have considerably narrowed the gap between lower and upper bounds, the gap for alphabets of intermediate size, i.e., $\omega(1) = k = n^{o(1)}$ remains to be closed and this is the second important open problem. For instance, for $k = \log_2 n$ the lower bound $\Omega(n \cdot (\log_2 \log_2 n)^2)$ and the upper bound $O(n \cdot \log_2 n \cdot \log_2 \log_2 n)$ are still by a factor of $\log_2 n / \log_2 \log_2 n$ apart.

Thirdly the size blowup when converting an NFA into an equivalent ε-free NFA remains to be determined. In [6] a family N_n of NFAs is constructed which has equivalent ε-free NFAs of size $\Omega(n^2 / \log_2^2 n)$ only. However the alphabet of N_n has size $n / \log_2 n$ and the gap between this lower bound and the corresponding upper bound $O(n^2 \cdot |\Sigma|)$ remains considerable.

Acknowledgement: Thanks to Gregor Gramlich and Juraj Hromkovic for many helpful discussions.

References

1. R. Book, S. Even, S. Greibach, G. Ott, Ambiguity in graphs and expressions, *IEEE Trans. Comput.* 20, pp. 149-153, 1971.
2. V. Geffert, Translation of binary regular expressions into nondeterministic ε-free automata with $O(n \log n)$ transitions, *J. Comput. Syst. Sci.* 66, pp. 451-472, 2003.
3. V.M. Glushkov, The abstract theory of automata, *Russian Math. Surveys* 16, pp. 1-53, 1961. Translation by J. M. Jackson from *Usp. Mat. Naut.* 16, pp. 3-41, 1961.
4. C. Hagenah, A. Muscholl, Computing ϵ-free NFA from regular expressions in $O(n \log^2(n))$ Time, *ITA*, 34 (4), pp. 257-278, 2000.
5. J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 2001.
6. J. Hromkovič, G. Schnitger, Comparing the size of NFAs with and without ε-transitions. *Theor. Comput. Sci.*, 380 (1-2), pp. 100-114, 2007.
7. J. Hromkovič, S. Seibert, T. Wilke, Translating regular expression into small ε-free nondeterministic automata, *J. Comput. Syst. Sci.*, 62, pp. 565-588, 2001.
8. Y. Lifshits, A lower bound on the size of ε-free NFA corresponding to a regular expression, *Inf. Process. Lett.* 85(6), pp. 293-299, 2003.
9. M.O. Rabin, D.Scott, Finite automata and their decision problems, *IBM J. Res. Develop.* 3, pp. 114-125, 1959.
10. S.Sippu, E. Soisalon-Soininen, Parsing Theory, Vol. I: Languages and Parsing, Springer-Verlag, 1988.
11. G. Schnitger, Regular expressions and NFAs without ε transitions, *Proc. of the 23rd STACS*, Lecture Notes in Computer Science 3884, pp. 432-443, 2006.
12. K. Thompson, Regular expression search, *Commun. ACM* 11, pp. 419-422, 1968.