Graphs Encoded by Regular Expressions

Stefan Gulan

Universität Trier, FB IV — Informatik
gulan@uni-trier.de

Abstract

In the conversion of finite automata to regular expressions, an exponential blowup in size can generally not be avoided. This is due to graph-structural properties of automata which cannot be directly encoded by regular expressions and cause the blowup combinatorially. In order to identify these structures, we generalize the class of arc-series-parallel digraphs to the acyclic case. The resulting digraphs are shown to be reversibly encoded by linear-sized regular expressions. We further derive a characterization of our new class by a finite set of forbidden minors and argue that these minors constitute the primitives causing the blowup in the conversion from automata to expressions.

1998 ACM Subject Classification F.4.3: Formal Languages, G.2.2: Graph Theory

Keywords and phrases Digraphs, Regular Expressions, Finite Automata, Forbidden Minors

Digital Object Identifier 10.4230/LIPIcs.STACS.2011.495

1 Motivation

A fundamental result in the theory of regular languages is the equivalent descriptive power of regular expressions and finite automata, as originally shown by Kleene [9]. While regular expressions come natural to humans as a means of denoting such languages, automata are the objects of choice on the machine level. Consequently, converting between these two representations is of great practical importance. There are several linear-time algorithms to transform regular expressions into automata with size linear in that of the input, a detailed overview is given by Watson [16]. We shall focus on the converse construction which is considerably more troubling.

In particular, Ehrenfeucht & Zeiger [3] give a class of automata for which the size of any equivalent expression is exponential in that of a given automaton. These automata are defined over an alphabet which grows with automaton size, which led Ellul et al. to ask whether a similar blowup in expression size can be shown for automata over a fixed alphabet [4]. An affirmative answer was given by Gruber & Holzer [5] for binary alphabets already. This mostly rules out alphabet size as a factor contributing to the exponential blowup, the modifier 'mostly' giving credit to the fact that automata over unary alphabets can be converted to expressions of quadratic size via Chrobak normal form [1, 4].

Observe that a finite automaton is merely a digraph with edge labels, accepting the language which consists of all sequences of labels met on a directed walk from an initial to a final state. Informally, the increase of expression- over automaton-size results from automata being combinatorial objects, whereas expressions are terms, i.e., linear entities, that must resort to repeated subterms in order to convey information encoded in the graph-structure of an automaton. This was observed quite early by McNaughton [11], who remarks that "although every regular expression can be transformed into a graph that has the same structure, the converse is not true". The present work aims to identify the graphs that cannot be transformed into expressions that have the same structure.

28th Symposium on Theoretical Aspects of Computer Science (STACS'11).
Editors: Thomas Schwentick, Christoph Dürr; pp. 495–506
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

For automata whose underlying graphs are arc-series-parallel, Moreira & Reis [12] recently gave an efficient conversion to expressions with size linear in that of the input. These graphs are characterized by absence of a single minor-like substructure in acyclic graphs, as was shown by Valdes et al. [15]. Korenblit & Levit [10] conjectured that this substructure already causes a quadratic blowup in the size of expressions constructed from acyclic automata.

However, Moreira & Reis's method is inherently confined to automata that accept finite languages only. In order to accept an infinite language, an automaton needs to contain cycles, which evades the class of arc-series-parallel digraphs. While separately dealing with series-parallel 'parts' of arbitrary automata has been suggested for conversion-heuristics [6], no strict graph-theoretic analysis has been conducted for the general case as yet.

This motivates our generalization of arc-series-parallel digraphs to the non-acyclic case in Sec. 3, yielding a class which is still efficiently recognizable. In Sec. 4 we show that such graphs can be reversibly encoded by regular expressions and that every regular expression encodes a graph of this class. Encoding and decoding is done in an automata-theoretic framework and can be immediately applied to the conversion between automata and expressions. In Sec. 5 we derive a characterization of our new class by a finite set of forbidden minors. This implies that these minors represent the graph-structural properties of automata that cannot be encoded by regular expressions and thus cause the blowup observed in the construction of regular expressions from finite automata.

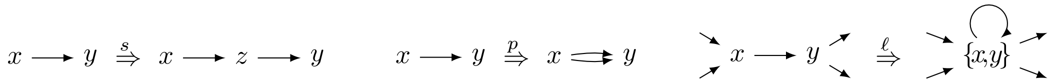
2 Preliminaries

We consider finite directed graphs with loops and multiple arcs. These are canonically known as *directed pseudographs* but will be referred to as just *graphs*. Formally, a graph is a tuple (V, A, t, h) with *vertices* V, *arcs* A, tail-map $t : A \to V$ and head-map $h : A \to V$. If G is not given explicitly, let $G = (V_G, A_G, t_G, h_G)$. An *xy-arc* of G is any $a \in A_G$ with $t_G(a) = x$ and $h_G(a) = y$; we write this as $a = xy \in A_G$. An xy-arc a *leaves* x and *enters* y, and x and y are called the *endpoints* of a. Distinct xy-arcs of a graph are *parallel* to each other. An xx-arc is an *x-loop* or just loop, every other arc is a *proper* arc. The *in-degree* of $x \in V_G$, denoted $\mathrm{d}^-_G(x)$, is the number of arcs entering x in G, the *out-degree* $\mathrm{d}^+_G(x)$ is the number of arcs leaving x. A *constriction* of G is any proper xy-arc where $\mathrm{d}^+_G(x) = 1 = \mathrm{d}^-_G(y)$. A vertex $x \in V_G$ is *simple* if $\mathrm{d}^-_G(x) \le 1$ and $\mathrm{d}^+_G(x) \le 1$. Subscripts are omitted if they are understood.

We write $F \subseteq G$ is F is a subgraph of G. If F and G are subgraphs of H and $a = xy \in A_H$ with $x \in V_F$ and $y \in V_G$, then a is called an (F, G)-arc, as well as an (x, G)- or an (F, y)-arc of H. A *path* of length n, denoted $\mathbf{P_n}$ is a graph on $n + 1$ vertices and n constrictions.

The *subdivision* of an arc $a = xy$ is the replacement of a with an xy-path of length 2. More generally, a subdivision of G, referred to as a DG, is any graph H s.t. there are graphs $G_1, \ldots, G_n$ where $G = G_1$, G_{i+1} results from subdividing an arc of G_i and $G_n = H$. The *split* of a vertex x is the replacement of x with two vertices x_1 and x_2 and an x_1x_2-arc and redirecting all arcs that entered x to enter x_1, resp. redirecting all arcs that left x to leave x_2. Two vertices $x, y \in V_G$ are *merged* by being replaced with a new vertex z and redirecting all arcs entering or leaving x or y to enter or leave z.

A graph G is *two-terminal* if there are $s, t \in V_G$ s.t. every $x \in V_G$ lies on some st-path in G. The vertices s and t are respectively called the *source* and *sink* of G; we write $G = (G, s, t)$ to express that G is two-terminal with source s and sink t. A two-terminal graph (G, s, t) is a *hammock* if $\mathrm{d}^-_G(s) = \mathrm{d}^+_G(t) = 0$. Let x and y be vertices of (G, s, t): x *dominates* y if x lies on every sy-path, and x *co-dominates* y if x lies on every yt-path. Furthermore, x is a *guard* of y if x dominates and co-dominates y; also, x is a guard of the arc a if x guards



(a) series expansion **(b)** parallel expansion **(c)** loop expansion

Figure 1 Expansions of an xy-arc, resp. the containing graph.

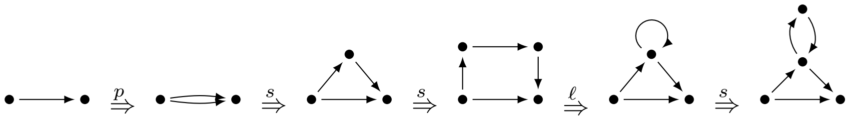


Figure 2 Construction of an spl-graph from $\mathbf{P_1}$ by a sequence of expansions.

both $t(a)$ and $h(a)$. More generally, x guards a subgraph F of (G, s, t) if x guards every arc and vertex of F.

3 SPL - Graphs

▶ **Definition 1.** The relations $\overset{s}{\Rightarrow}$, $\overset{p}{\Rightarrow}$ and $\overset{\ell}{\Rightarrow}$ are defined on graphs as follows: Let G be a graph and a an xy-arc in G, then

- $G \overset{s}{\Rightarrow} H$ if H is obtained by subdividing a in G
- $G \overset{p}{\Rightarrow} H$ if H is obtained from G by adding an arc which is parallel to a.
- $G \overset{\ell}{\Rightarrow} H$ if a is a constriction and H is obtained by merging x and y in G.

We say that H is derived from G by means of *series*-, *parallel*- or *loop*-expansion if $G \overset{s}{\Rightarrow} H$, $G \overset{p}{\Rightarrow} H$ or $G \overset{\ell}{\Rightarrow} H$, respectively. The local changes in G upon expansion are sketched in Fig. 1. We write $G \Rightarrow H$ if the particular expansion is irrelevant, and $G \Rightarrow^\star H$ if H is derived from G by a (possibly empty) finite sequence of expansions.

▶ **Definition 2.** The class of spl-graphs, denoted $\mathcal{SPL}$, is generated by $\Rightarrow$ from $\mathbf{P_1}$ as follows

- $\mathbf{P_1} \in \mathcal{SPL}$
- Let $G \in \mathcal{SPL}$, then $H \in \mathcal{SPL}$ if $G \overset{s}{\Rightarrow} H$ or $G \overset{p}{\Rightarrow} H$, or if $G \overset{\ell}{\Rightarrow} H$ where the ℓ-expanded arc is not incident to the source or the sink of G.

We call $\mathbf{P_1}$ the *axiom* of $\mathcal{SPL}$. The restriction imposed on ℓ-expansion ensures that every spl-graph is a hammock. An example for the step-wise construction of an spl-graph is shown in Fig. 2. The acyclic spl-graphs coincide with the arc-series-parallel graphs investigated by Valdes et al. [15]; we resort to their results whenever possible and elaborate only on properties of $\mathcal{SPL}$ that arise from its non-acyclic members.

To decide whether (G, s, t) is an spl-graph, we define a kind of dual to expansion. Some care must be taken with the removal of loops, which is why the new operations are restricted to hammocks.

▶ **Definition 3.** The relations $\overset{s}{\Leftarrow}$, $\overset{p}{\Leftarrow}$ and $\overset{\ell}{\Leftarrow}$ are defined on hammocks as follows: Let $G = (G, s, t)$ be a hammock, then

i) $G \overset{s}{\Leftarrow} H$ if y is simple vertex of G, incident to $a_1 = xy$ and $a_2 = yz$, and H is derived from G by removing y, a_1 and a_2 and adding an xz-arc.
ii) $G \overset{p}{\Leftarrow} H$ if H is derived from G by removing one of two parallel arcs.

iii) $G \overset{\ell}{\Leftarrow} H$ if a is an x-loop in G s.t. x does not guard any arc besides a, and H is the split of x in $G \setminus \{a\}$.

If $G \overset{c}{\Leftarrow} H$ for $c \in \{s, p, \ell\}$ we call both H and the replacement operation a c-*reduction* of G. As before we may simply write $G \Leftarrow H$ for any reduction, and $G \Leftarrow^\star H$ if H can be derived from G by a sequence of reductions.

Expansion and reduction are not proper duals as the latter relation is restricted to hammocks by definition. For hammocks we find $G \overset{c}{\Leftarrow} H$ *iff* $H \overset{c}{\Leftarrow} G$ for $c \in \{s, p\}$; but while $G \overset{\ell}{\Leftarrow} H$ implies $H \overset{\ell}{\Rightarrow} G$, the converse is not true. The asymmetry is due to the fact that if ℓ-expansion introduces an x-loop a, x might guard some arc besides a, so the converse reduction is not ensured. This, however, does not happen within $\mathcal{SPL}$.

▶ **Proposition 4.** *Let C be a cycle of $G \in \mathcal{SPL}$. Then exactly one vertex of C guards C.*

The intuition of Prop. 4 is that every cycle in an spl-graph contains a vertex that serves as the unique 'entry' and 'exit' of this cycle wrt. the source and sink (see the last two steps in Fig. 2). Also note that a cycle might well be guarded by any number of vertices outside the cycle.

▶ **Theorem 5.** $G \in \mathcal{SPL}$ iff $G \Leftarrow^\star \mathbf{P_1}$

Proof. Since $G \Leftarrow^\star \mathbf{P_1}$ implies $\mathbf{P_1} \Rightarrow^\star G$, reducibility is sufficient for membership. Necessity is shown by induction on the structure of G. The claim holds for $\mathbf{P_1}$, so suppose $G \in \mathcal{SPL}$ where $G \Leftarrow^\star \mathbf{P_1}$ and let $G \Rightarrow H$. We attend to ℓ-expansion only, the other cases are trivial. Let $a = uv$ be the relevant constriction of G and let $l = xx$ be the loop introduced in H. If x guards some distinct arc $a' = yz$ in H, then G contains a cycle that defies Prop. 4, contrary to the assumption $G \in \mathcal{SPL}$. Therefore, $H \overset{\ell}{\Leftarrow} G$ is a valid reduction; since by assumption $G \Leftarrow^\star \mathbf{P_1}$, we find $H \Leftarrow^\star \mathbf{P_1}$. ◀

While membership in $\mathcal{SPL}$ can be decided by reducing a hammock to the axiom of $\mathcal{SPL}$, we do not know how to do so. Actually, there is no need for a strategy, since the reduction-system exhibits unique normal-forms. Using a standard argument from abstract rewriting (see e.g. [14]), we first show that reductions are locally confluent.

▶ **Lemma 6.** *Let G be a hammock and suppose $G \Leftarrow H_1$ and $G \Leftarrow H_2$ hold. Then there is a hammock J s.t. $H_1 \Leftarrow^\star J$ and $H_2 \Leftarrow^\star J$ hold.*

Each reduction decreases the number of arcs or loops and none introduces loops, so every sequence of reductions eventually terminates. Any graph derived from G by exhaustive reduction is called *normal-form* of G and denoted $\mathrm{R}(G)$. A graph G that coincides with its normal-form, $G = \mathrm{R}(G)$, is called *reduced*. Applying Newman's lemma [13, 14] yields

▶ **Corollary 7.** *The normal-form $\mathrm{R}(G)$ of any hammock G is unique.*

Computing the normal-form of a hammock can be realized by repeatedly running the reduction algorithm for arc-series-parallel graphs [15], interspersed with ℓ-reductions, until no further reduction can be applied. To this end some bookmarking about the loops occurring in the intermediate graphs is necessary. Testing whether x is a guard can be done in linear time by counting the components of $G \setminus x$. Overall, this method computes $\mathrm{R}(G)$ in quadratic time.

▶ **Theorem 8.** *Membership of G in $\mathcal{SPL}$ is effectively decidable due to*

$$G \in \mathcal{SPL} \quad \text{iff} \quad G \text{ is a hammock and } \mathrm{R}(G) = \mathbf{P_1}$$

4 Encoding by Regular Expressions

Syntax and semantics of regular expressions (REs) follow Hopcroft & Ullman's textbook [8] except that we do not allow for $\emptyset$ in REs. As for notation, L_r denotes the language described by the RE r and $\mathrm{reg}(\Sigma)$ denotes the class of REs over Σ. An RE is *simplified* if it does not contain ε as a factor. Any RE r can be converted to a simplified RE $\mathrm{simp}(r)$, denoting the same language, by replacing every subexpression $s\varepsilon$ or εs with just s.

An *extended finite automaton* (EFA) over Σ is a 5-tuple $E = (Q, \Sigma, \delta, I, F)$, whose elements denote the set of states, the alphabet, the transition relation, the initial and the final states, respectively. These sets are all finite and satisfy $Q \cap \Sigma = \emptyset$, $\delta \subseteq Q \times \mathrm{reg}(\Sigma) \times Q$, $I \subseteq Q$, and $F \subseteq Q$. The relation $\vdash_E$ is defined on $Q \times \Sigma^*$ as $(q, ww') \vdash_E (q', w')$ if $(q, r, q') \in \delta$ and $w \in L_r$. The language accepted by E is

$$L(E) := \{w \mid (q_i, w) \vdash_E^* (q_f, \varepsilon) \text{ for } q_i \in I,\, q_f \in F\}$$

Two EFAs are *equivalent* if they accept the same language. An EFA is *normalized* if $|I| = |F| = 1$ and the initial and final state are distinct; any EFA can normalized by adding a new initial (final) state and ε-transitions from (to) this new initial (final) state to (from) the original ones. The EFA E is *trim* if for every state q of E there is a word $w = w_1w_2 \in L(E)$ s.t. $(q_i, w_1) \vdash_E^* (q, \varepsilon)$ and $(q, w_2) \vdash_E^* (q_f, \varepsilon)$ hold for some $q_i \in I$ and $q_F \in F$. Any EFA can be converted to a trim equivalent EFA by removing all states that do not meet this requirement and adjusting the transition relation. A *nondeterministic finite automaton* with ε-transitions (εNFA) is an EFA whose transition relation is restricted to $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$.

The graph *underlying* E is $G(E) := (Q, \delta, t, h)$ where $t : (p, r, q) \mapsto p$ and $h : (p, r, q) \mapsto q$. It is easy to see that E is trim and normalized *iff* $G(E)$ is a hammock.

An EFA displays a compromise between the complexity of its transition-labels and that of its underlying graph; REs and εNFAs represent the extremes in this tradeoff: an RE can be considered as an EFA whose underlying graph is trivial, namely $\mathbf{P_1}$, while an εNFA is an EFA with trivial labels. Locally relaying information about a language between the graph-structure of an EFA and its labels lies at the heart of several conversions between REs and εNFAs.

4.1 Expressions to Automata

We consider a fragment of the replacement-system proposed by Gulan & Fernau [7]. Let E be an EFA with transition $\tau = (p, r, q)$ where r contains operators, then τ can be replaced depending on the root of r, the out-degree of p and the in-degree of q. The degrees are only relevant if r is an iteration: in this case they determine whether p and q should be merged or a new state should be added (or neither) upon introduction of a loop. The rewriting rules, denoted $\triangleleft_\bullet$, $\triangleleft_+$, and $\triangleleft_{*1}$ to $\triangleleft_{*4}$ are shown in Fig. 3.

In order to convert an RE into an εNFA, we identify $r \in \mathrm{reg}(\Sigma)$ with the trivial EFA $A_r^0 := (\{q_i, q_f\}, \Sigma, \{(q_i, r, q_f)\}, \{q_i\}, \{q_f\})$, which obviously satisfies $L(A_r^0) = L_r$. The language accepted by an EFA is invariant under each rewriting, hence exhaustive application of $\triangleleft_\bullet$, $\triangleleft_+$ and $\triangleleft_{*i}$ yields a sequence $A_r^0, A_r^1, \ldots$ of equivalent EFAs terminating in an εNFA which we denote A_r.

▶ **Lemma 9.** *Every A_r^i satisfies $G(A_r^i) \in \mathcal{SPL}$.*

Thus the graph underlying A_r is an spl-graph, too. It is shown in [7] that A_r is unique. We thus define a map α from $\mathrm{reg}(\Sigma)$ to the class of expression-labeled spl-graphs (aka trim normalized EFAs) by setting $\alpha(r) := A_r$.

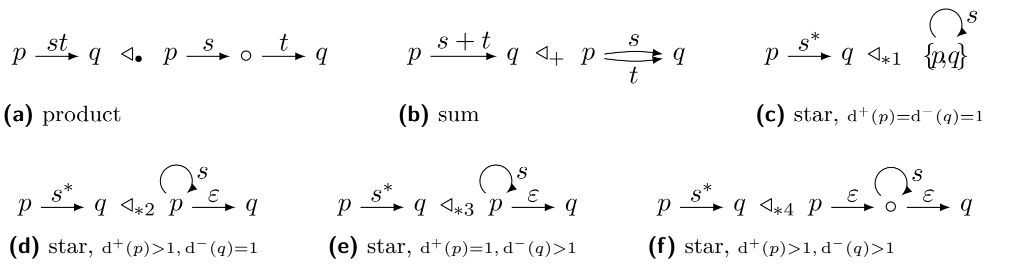


Figure 3 Replacing a transition (p, r, q) based on its label and, in case $r = s^*$, the out-degree of p and the in-degree of q in $G(E)$. Either rule $\lhd_\bullet$, $\lhd_{*4}$ introduces a new state 'between' p and q.

4.2 Automata to Expressions

The spl-reductions are augmented to handle expression-labeled arcs, which yields a second rewriting-system on EFAs. In order to meet the requirements for loop-reduction, we consider normalized EFAs only. The labeled reductions, denoted $\rhd_\bullet$, $\rhd_+$, and $\rhd_*$, are shown in Fig. 4. Again, the accepted language is invariant under these transformations.

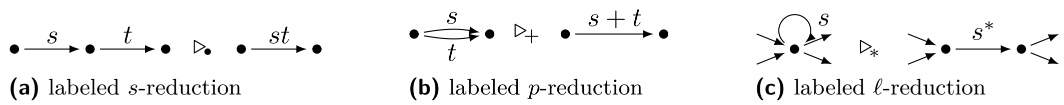


Figure 4 Labeled spl-reductions

Exhaustive reduction of a normalized EFA E terminates in an equivalent EFA which we denote $\mathrm{R}_l(E)$. The graph underlying $\mathrm{R}_l(E)$ is the normal-form of the graph underlying E, $G(\mathrm{R}_l(E)) = \mathrm{R}(G(E))$, and we further find

▶ **Proposition 10.** *The labels of* $\mathrm{R}_l(E)$ *are unique up to associativity and commutativity.*

In particular if $G(E) \in \mathcal{SPL}$, we find $G(\mathrm{R}_l(E)) = \mathbf{P_1}$, so the only label of $\mathrm{R}_l(E)$ is an RE r with $L_r = L(E)$. By Prop. 10 this RE is unique up to trivialities, so we define a map β from EFAs with spl-structure to REs by setting $\beta(E) := r$, where r is the label of $\mathrm{R}_l(E)$.

4.3 Duality of the Conversions

The conversions between REs and εNFAs with spl-structure are 'almost' duals, some extra effort arises with the treatment of ε-factors resp. certain ε-labeled transitions. This is due to the fact that star-expansion might introduce ε-transition that have no corresponding subterm in the

We write $r = r'$ if the expressions r and r' are identical up to associativity and commutativity of the regular operators.

▶ **Theorem 11.**

1. $\mathrm{simp}(r) = \beta(\alpha(\mathrm{simp}(r)))$ *for any RE* r
2. $A = \alpha(\mathrm{simp}(\beta(A)))$ *for any* ε*NFA* A *with* $G(A) \in \mathcal{SPL}$

Thus the encoding of labeled spl-graphs by simplified expressions is unique and reversible, and every simplified expression encodes a labeled spl-graph. Hence every RE over a non-empty alphabet encodes an spl-graph and every spl-graph can be encoded. Informally, we state

▶ **Corollary 12.** *$G \in \mathcal{SPL}$ iff G can be encoded by an RE*

5 Forbidden Minor Characterization

We adapt the notion of topological minors, which is well-known for undirected graphs (see e.g. [2]), to our needs.

▶ **Definition 13.** An *embedding* of F in G is an injection $e : V_F \to V_G$ satisfying that if $a = xy \in A_F$, then G contains an $e(x)e(y)$-path P_a, and that P_a and $P_{a'}$ are internally disjoint for distinct $a, a' \in A_F$.

If an embedding of F in G exists, we call F a *minor* of G *realized* by the embedding. We write $F \preccurlyeq G$ if F is a minor of G. If $F \preccurlyeq G$ does not hold then G is F*-free*; if $\mathcal{M}$ is a set of graphs and G is F-free for every $F \in \mathcal{M}$, then G is $\mathcal{M}$-free. It is easily seen that subdivisions allow for an equivalent characterization of minors:

▶ **Proposition 14.** *$F \preccurlyeq G$ iff G contains a* DF

Let $F \preccurlyeq G$ be realized by e and $x \in V_F$, we call $e(x)$ a *peg* of F in G wrt. e; if G and e are known, we omit mentioning them. Observe that the in-/out-degree of a vertex in F does not exceed the in-/out-degree of its corresponding peg in G:

▶ **Proposition 15.** *If e realizes $F \preccurlyeq G$, then $\mathrm{d}^-_F(x) \leq \mathrm{d}^-_G(e(x))$ and $\mathrm{d}^+_F(x) \leq \mathrm{d}^+_G(e(x))$.*

Let e realize $F \preccurlyeq G$, a *bypass* of F in G wrt. e is an $e(x)e(y)$-path in G, where xy is *not* an arc of F. An embedding of F in G is *bare* if G contains no bypass of F wrt. to the embedding; we then write $M \sqsubseteq G$. Observe that $F \preccurlyeq G$ might well be realized by various — in particular bare and non-bare — embeddings. Based on Prop. 14, we also call a DF in G bare if G contains no bypass wrt. to the embedding realizing this DF.

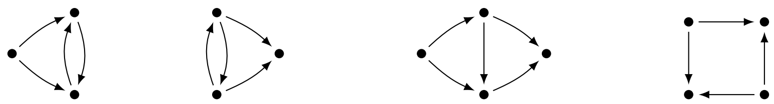
The existence of an xy-path is invariant under spl-expansion and -reduction if x and y are not subject to the operation.

▶ **Proposition 16.** *Let $G \Rightarrow H$ or $G \Leftarrow H$ and $\{x, y\} \subseteq V_G \cap V_H$, then G contains an xy-path* iff *H does.*

'Half' of the sought characterization is given by the set of graphs $\boldsymbol{\mathcal{F}} = \{\mathbf{C}, \mathbf{C}^{\mathrm{R}}, \mathbf{N}, \mathbf{Q}\}$, shown in Fig. 5. Note that Valdes et al. proved that an acyclic hammock is arc-series-parallel *iff* it is $\mathbf{N}$-free [15].

▶ **Lemma 17.** *Every $G \in \mathcal{SPL}$ is $\boldsymbol{\mathcal{F}}$-free.*

Proof. Clearly, $\mathbf{P_1}$ is $\boldsymbol{\mathcal{F}}$-free. Assume $G \in \mathcal{SPL}$ is $\boldsymbol{\mathcal{F}}$-free and let $G \Rightarrow H$. Consider any $F \in \boldsymbol{\mathcal{F}}$: since F is free of parallel arcs, and the existence of paths among vertices in $V_G \cap V_H$ is invariant under expansion (Prop. 16), $F \preccurlyeq H$ implies that a peg of F in H was introduced upon expansion. Hence in case $G \overset{p}{\Rightarrow} H$, F is not a minor of H, i.e., H is $\boldsymbol{\mathcal{F}}$-free. The same goes for $G \overset{s}{\Rightarrow} H$: as the new vertex in H is simple, but no vertex of F is, Prop. 15 implies that H is F-free and therefore $\boldsymbol{\mathcal{F}}$-free.
If $G \overset{\ell}{\Rightarrow} H$, let $a = xy$ be the relevant constriction of G and $l = zz$ the loop of H introduced by expansion. If $F \preccurlyeq H$ is realized by e, then $z = e(q)$ for some $q \in V_F$, as was discussed above. Let $H' = H \setminus l$: since F is free of loops, $F \preccurlyeq H'$ holds, too, and since q is not simple in F, z is not simple in H'. We actually find $\mathrm{d}^-_{H'}(z) \geq 2$ *and* $\mathrm{d}^+_{H'}(z) \geq 2$: if $\mathrm{d}^-_{H'}(z) = 0$, then $F \preccurlyeq G$ is realized by e', which is defined as e except that $e'(q) = y$ — contradicting the assumption that G is $\boldsymbol{\mathcal{F}}$-free. If $\mathrm{d}^-_{H'}(z) = 1$, there is exactly one arc entering z in H. Let



(a) $\mathbf{C}$ **(b)** $\mathbf{C}^{\mathrm{R}}$ **(c)** $\mathbf{N}$ **(d)** $\mathbf{Q}$

Figure 5 Graphs constituting $\boldsymbol{\mathcal{F}}$.

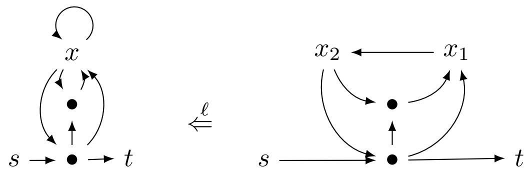


Figure 6 The graph $\mathbf{N}$ emerges as a subgraph due to ℓ-reduction of a hammock. Still, $\mathbf{C}$ is a minor of either side.

this be $a' = z'z$, then $F \preccurlyeq G$ is realized by e'' which is as e except that, again, $e''(q) = y$, contradicting our assumption. A symmetric argument shows $\mathrm{d}^+_{H'}(z) \geq 2$. In fact, we have also shown that q, of which z is the peg, has in- and out-degree at least two.
But since $\mathrm{d}^-_G(x) = \mathrm{d}^-_{H'}(z)$ and $\mathrm{d}^+_G(y) = \mathrm{d}^+_{H'}(z)$, some $F' \in \boldsymbol{\mathcal{F}}$, constructed by splitting q in F satisfies $F' \preccurlyeq G$ — contradicting the assumption that G is $\boldsymbol{\mathcal{F}}$-free. ◀

Likewise, it can be shown in general that if $H \Leftarrow G$ and H is not $\boldsymbol{\mathcal{F}}$-free, then neither is G. However, there is a catch: the $\boldsymbol{\mathcal{F}}$-minors of G and H need not coincide. This is hinted at by the following lemma, and an explicit example is shown in Fig. 6.

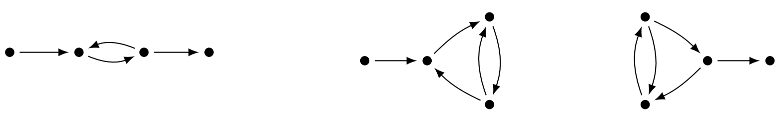
▶ **Lemma 18.** *If $H \Leftarrow G$ for hammocks H and G, then H is $\boldsymbol{\mathcal{F}}$-free* iff *G is. More specifically:*

i) $F \preccurlyeq H$ iff $F \preccurlyeq G$ *for* $F \in \{\mathbf{C}, \mathbf{C}^{\mathrm{R}}, \mathbf{Q}\}$
ii) $\mathbf{N} \preccurlyeq H$ *only if* $\mathbf{N} \preccurlyeq G$*, whereas*
iii) $\mathbf{N} \preccurlyeq G$ *only if* $(\mathbf{N} \preccurlyeq G$ *or* $\mathbf{C} \preccurlyeq G$ *or* $\mathbf{C}^{\mathrm{R}} \preccurlyeq G)$

Still, $\boldsymbol{\mathcal{F}}$-freeness of a hammock is not sufficient for membership in $\mathcal{SPL}$: for example, the hammock $\mathbf{\Phi}$, shown in Fig. 7a, is $\boldsymbol{\mathcal{F}}$-free, but not included in $\mathcal{SPL}$. The additional graphs necessary for a characterization by forbidden minors are $\mathbf{\Phi}$, $\mathbf{\Psi}$, and $\mathbf{\Psi}^{\mathrm{R}}$, shown in Fig. 7.

▶ **Lemma 19.** *Every $G \in \mathcal{SPL}$ is free of bare $\mathbf{\Phi}$-, $\mathbf{\Psi}$-, and $\mathbf{\Psi}^{\mathrm{R}}$-minors*

On the other hand, each of $\{\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Psi}^{\mathrm{R}}\}$ may well be a minor of certain spl-graphs. Examples for spl-graphs with $\mathbf{\Phi}$ and $\mathbf{\Psi}$ as minors are given in Fig. 8, the reader might want to check that these can be reduced to $\mathbf{P_1}$. In the absence of $\boldsymbol{\mathcal{F}}$-minors an invariance-result akin to Lem. 18 holds for bare subdivisions of these three graphs.



(a) $\mathbf{\Phi}$ **(b)** $\mathbf{\Psi}$ **(c)** $\mathbf{\Psi}^{\mathrm{R}}$

Figure 7 Graphs that do not allow for a bare embedding in any $G \in \mathcal{SPL}$.

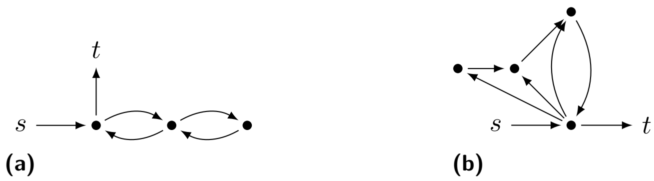


Figure 8 Examples for spl-graphs with $\mathbf{\Phi}$ and $\mathbf{\Psi}$ as (non-bare) minors

▶ **Lemma 20.** *Let H be an $\boldsymbol{\mathcal{F}}$-free hammock and assume $H \Leftarrow G$, then $F \sqsubseteq H$ iff $F \sqsubseteq G$ for $F \in \{\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Psi}^{\mathrm{R}}\}$.*

Proof. Each of $\mathbf{\Phi}$, $\mathbf{\Psi}$, and $\mathbf{\Psi}^{\mathrm{R}}$ is free of parallel arcs, so Prop. 16 yields the claim if all pegs occur in $V_G \cap V_H$; in particular, nothing needs to be done for $H \overset{p}{\Leftarrow} G$. We prove the claim for $\mathbf{\Phi}$, the procedure is the same for $\mathbf{\Psi}$ and $\mathbf{\Psi}^{\mathrm{R}}$. In the following, let H be $\boldsymbol{\mathcal{F}}$-free.
Let $\mathbf{\Phi} \sqsubseteq H$ be realized by e. If $G \overset{s}{\Leftarrow} H$ removes a peg $x = e(q)$, q is one of the two simple vertices of $\mathbf{\Phi}$; here, let q be the unique vertex with $\mathrm{d}^-_{\mathbf{\Phi}}(q) = 0$ (the other case is symmetric). Since s-reduction is applicable due to x, an arc $a = yx$ exists in H, with y also occurring in G. Let e' be an embedding of $\mathbf{\Phi}$ in G, s.t. $e'(q) = y$ and e' as e for the other vertices. If e' is bare, the claim follows for $\mathbf{\Phi}$ and s-reduction, so assume it is not. Then G contains a bypass of $\mathbf{\Phi}$ wrt. e', which is necessarily a path *leaving* y, otherwise H would contain a bypass of $\mathbf{\Phi}$ wrt. e, contradicting the assumption that e is bare. We find $\mathbf{C} \preccurlyeq G$, if the other endpoint of the bypass is the peg of the vertex in $\mathbf{\Phi}$'s cycle that is not adjacent to q. If the bypass is from $e'(q)$ to the peg of the vertex with out-degree 0 in $\mathbf{\Phi}$, we get $\mathbf{Q} \preccurlyeq G$. In both cases Lem. 18 implies that H is not $\boldsymbol{\mathcal{F}}$-free, contradicting our assumption. Proving that s-reduction does not introduce new bare D$\mathbf{\Phi}$'s is trivial.
Again let $\mathbf{\Phi} \sqsubseteq H$ be realized by e with peg $x \in V_H$. Considering $H \overset{\ell}{\Leftarrow} G$, let $a = xx$ be the loop that allows for reduction, and let x_1x_2 denote that constriction arising from it. As in the proof of Lem. 18 our argument is based on the facts that a is irrelevant for the D$\mathbf{\Phi}$ in H and that $\mathrm{d}^-_G(x_1) = \mathrm{d}^-_{H\setminus a}(x)$ and $\mathrm{d}^+_G(x_2) = \mathrm{d}^-_{H\setminus a}(x)$ hold. Since every of $\mathbf{\Phi}$ has either in- or out-degree ≤ 1, we can construct an embedding e' of $\mathbf{\Phi}$ in G by assigning the role of x to either x_1 or x_2. ◀

▶ **Definition 21.** A *kebab* is a connected graph consisting of three arc-disjoint subgraphs: a strong component B, called the *body*, and two nonempty vertex-disjoint paths S_1 and S_2, called the *spikes* of the kebab.

We name some unique vertices in a kebab: the endpoint of a spike connecting that spike to the body is the *puncture* of this spike, the other endpoint is its *tip*. A spike which enters the body of a kebab is an *in-spike*, one that leaves the body is called an *out-spike*. If both spikes of a kebab K enter (leave) the body, K is also called an *in*-kebab (*out*-kebab) if one enters and the other leaves the body, K is called an *inout*-kebab. In order to prove two lemmas concerning kebabs, some auxilliary propositions are necessary.

▶ **Proposition 22.** *Let G be a reduced hammock with distinct arcs a_1, a_2 s.t. $h(a_1) = v = t(a_2)$. Then v is incident to a third proper arc.*

▶ **Proposition 23.** *Let x and y be distinct vertices of a hammock G. Then exactly one of the following is true: 1) x dominates y, 2) y dominates x, or 3) for some $z \in V_G \setminus \{x, y\}$, G contains internally disjoint zx- and zy-paths.*

▶ **Proposition 24.** *Let x and y be distinct vertices of a strong graph G, then there is a cycle $C \subseteq G$ and distinct $z_x, z_y \in V_C$, s.t. G contains an xz_x- and a yz_y-path that are disjoint.*

▶ **Lemma 25.** *Let (G, s, t) be an spl-reduced hammock and suppose G contains a kebab. Then $F \preccurlyeq G$ for some $F \in \boldsymbol{\mathcal{F}}$ or $\mathbf{\Phi} \sqsubseteq G$.*

Proof. We choose a 'biggest' kebab $K \subseteq G$ with the following properties

1. the body of K is arc-maximal, i.e., no kebab of G has a body with more arcs
2. the spikes of K are inclusion-maximal in G, i.e., they are not 'sub-spikes' of a bigger kebab with the same body as K but longer spikes than K.

We need to distinguish whether K is in an in-, an out- or an inout-kebab. Due to space restrictions we only treat the first case, however note that the first and second case are symmetric.

Let $K \subseteq G$ be an in-kebab and let B denote the body of K, S_1 and S_2 the spikes, with tips t_1 and t_2, and punctures p_1 and p_2, respectively (Fig. 9a). As (G, s, t) is a hammock, according to Prop. 23 either one of t_1 and t_2 dominates the other, or G contains a vertex x and internally disjoint xt_1- and xt_2-paths.

1. If t_2 dominates t_1 (the converse case is symmetric), let P be a shortest t_2t_1-path in G. If P and B are disjoint, then P contains a segment P' from S_2 to S_1. Using Prop. 24 we now find $\mathbf{C} \preccurlyeq G$ (Fig. 9b). So let P go through B, then the last segment of P is a (B, t_2)-path outside B. By the choice of K and P this segment consist of a single arc $a = bt_1$ for $b \in V_B$ (Fig. 9c). According to Prop. 22, t_1 is incident to a further arc a', as G is reduced. Our choice of K requires that the other endpoint z of a' lies in K, since B, S_1 and a form a strong component bigger than B. It is now easy to see (from Fig. 9c), that $z \in V_{S_2}$ yields $\mathbf{C} \preccurlyeq G$ (regardless of a's orientation), and that $z \in V_B$ yields $\mathbf{C} \preccurlyeq G$ or $\mathbf{C}^{\mathrm{R}} \preccurlyeq G$ (depending on a's orientation), so let $z \in V_{S_1} \setminus \{p_1\}$. This leaves two possibilities: If $a' = zt_1$ we find $\mathbf{Q} \preccurlyeq G$, with pegs t_1,p_1,b and z (Fig. 9d). On the other hand, $a' = t_1z$ leads to a contradiction: Since G is p-reduced, there is at least one vertex z' between t_1 and z on S_1; omitting the t_1z'-segment of S_1 lets us identify an in-kebab with tips z' and t_2 and a body properly containing B (Fig. 9e), contradicting maximality of B.

2. Let G contain a zt_1-path P_1 and a zt_2-path P_2 which are internally disjoint. If both P_i are disjoint with B, we find $\mathbf{C} \preccurlyeq G$ with help of Prop. 24 (Fig. 9f, where x_i denotes the first vertex on P_i that is also in S_i). If wlog. P_1 intersects B, let b denote the last vertex on P_1 that is in B and x the first vertex on P_1 that is in $V_{S_i} \setminus \{p_i\}$. If $x \neq t_1$, we find a kebab in G with a body containing B, contradicting our choice of K. As the claim was already proven for $x = t_1$ (see Fig. 9c), the statement follows for in-kebabs.

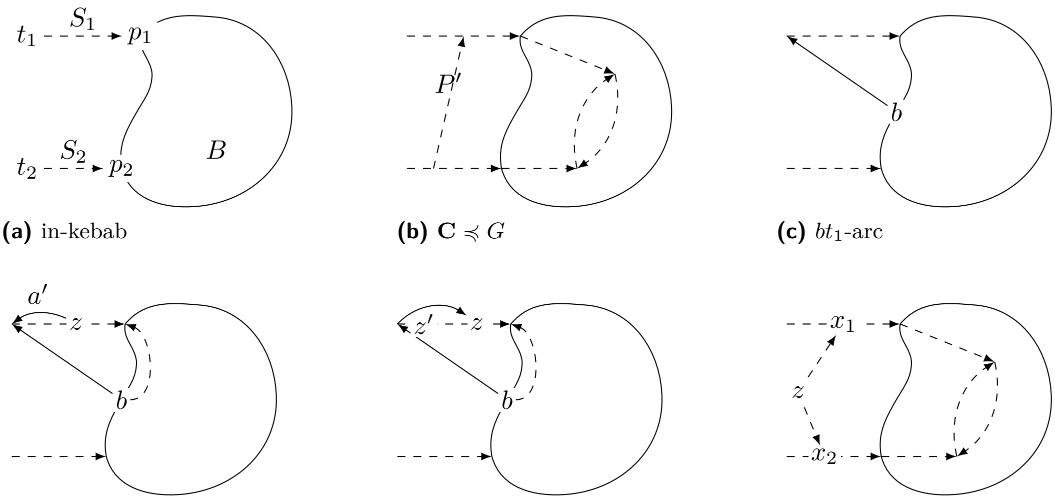
◀

▶ **Lemma 26.** *Let $G \neq \mathbf{P_1}$ be a reduced hammock with cycles. Then $F \preccurlyeq G$ for some $F \in \boldsymbol{\mathcal{F}}$ or $F' \sqsubseteq G$ for some $F' \in \{\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Psi}^{\mathrm{R}}\}$.*

We have thus found a characterization of $\mathcal{SPL}$ by forbidden subgraphs.

▶ **Theorem 27.** *Let G be a hammock, then*

$$G \in \mathcal{SPL} \quad \text{iff} \quad G \textit{ is } \boldsymbol{\mathcal{F}}\textit{-free and no } F' \in \{\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Psi}^{\mathrm{R}}\} \textit{ is a bare minor of } G.$$



(a) in-kebab **(b)** $\mathbf{C} \preccurlyeq G$ **(c)** bt_1-arc

(d) $\mathbf{Q} \preccurlyeq G$ **(e)** $a' = t_1 z$ **(f)** $\mathbf{C} \preccurlyeq G$

Figure 9 Cases occurring in the proof of Lem. 25 for K being an in-kebab. Solid arrows represent arcs, dashed arrows represent paths.

Proof. Let $G \in \mathcal{SPL}$, then Lem. 17 states that G is $\boldsymbol{\mathcal{F}}$-free, while Lem. 19 states that none of $\{\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Psi}^{\mathrm{R}}\}$ is a bare minor of G. Conversely if $G \notin \mathcal{SPL}$ then Cor. 7 yields $\mathrm{R}(G) \neq \mathbf{P_1}$. By Valdes' result and Lem. 26, we know $F \preccurlyeq \mathrm{R}(G)$ for some $F \in \boldsymbol{\mathcal{F}}$ and/or $F' \sqsubseteq \mathrm{R}(G)$ for some $F' \in \{\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Psi}^{\mathrm{R}}\}$. If $G = \mathrm{R}(G)$, i.e., G is already reduced, the claim follows immediately; otherwise, induction on the length of the reduction using Lems. 18 and 20 provides the statement. ◀

6 Conclusions

We generalized the class of arc-series-parallel graphs by augmenting the standard construction with a rule that allows for loops. Members of the new class can be reversibly encoded by regular expressions which represent the recursive structure of a graph; naturally, the size of such an encoding is linear in that of the input. Moreover, any regular expression represents an spl-graph under this encoding. Modulo isomorphism of graphs, resp. modulo associativity and commutativity of operators in expressions, the encoding and decoding are unique; thus they provide — up to trivialities — a bijection between spl-graphs and regular expressions.

The encoding is done by constructing a series of arc-labeled spl-graphs. As an automaton can be interpreted as an arc-labeled graph, this can be immediately applied to the conversion of finite automata with spl-structure to equivalent regular expressions whose size is linear wrt. to the automaton. This generalizes a recent result for acyclic automata.

We further characterized our new class by means of 7 forbidden minors. Therefore the exponential increase of expression size over automaton size, which cannot be avoided in the general case, is due to graph-structural properties of automata that are not present in spl-graphs. The forbidden minors can be considered as being the primitives of these non-expressible properties, they should be further investigated in order to improve on current conversions from automata to expressions.

References

1 Marek Chrobak. Finite automat and unary languages. *Theoretical Computer Science*, (47):149–158, 1986.

2 Reinhard Diestel. *Graph Theory.* Number 173 in Graduate Texts in Mathematics. Springer, 4 edition, 2010.

3 A. Ehrenfeucht and P. Zeiger. Complexity measures for regular expressions. *Journal of Computer and System Sciences*, 12:134–146, 1976.

4 K. Ellul, B. Krawetz, J. Shallit, and M.-W. Wang. Regular expressions: new results and open problems. *Journal of Automata, Languages and Combinatorics*, 10(4):407–437, 2005.

5 H. Gruber and M. Holzer. Finite automat, digraph connectivity, and regular expression size. In *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 39–50, 2008.

6 S. Gulan and H. Fernau. Local elimination-strategies in finite automata for shorter regular expressions. In *SOFSEM 2008*, volume 2, pages 46–57, 2008.

7 S. Gulan and H. Fernau. An optimal construction of finite automata from regular expressions. In R. Hariharan et al., editor, *28th FSTTCS*, pages 211–222, 2008.

8 John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

9 S. C. Kleene. *Representation of Events in Nerve Nets and Finite Automata*, pages 3–41. Annals of Mathematics Studies. 1956.

10 M. Korenblit and V. E. Levit. On algebraic expressions of series-parallel and fibonacci graphs. In C. S. Calude et al., editor, *DMTCS 2003*, number 2731 in LNCS, pages 215–224, 2003.

11 R. McNaughton. Techniques for manipulating regular expressions. In J. F. Hart and S. Takasu, editors, *Systems and Computer Science*, pages 27–41. University of Toronto Press in association with the University of Western Ontario Toronto, 1967.

12 Nelma Moreira and Rogério Reis. Series-parallel automata and short regular expressions. *Fundamenta Informaticae*, 91(3-4):611–629, 2009.

13 M.H.A. Newman. On theories with a combinatorial definition of ”equivalence”. *Annals of Mathematics*, 43(2), 1942.

14 Enno Ohlebusch. *Advanced Topics in Term Rewriting.* Springer, 2002.

15 Jacobo Valdes, Robert E. Tarjan, and Eugene L. Lawler. The recognition of series parallel digraphs. *SIAM Journal on Computing*, 11(2):298–313, 1981.

16 B. W. Watson. A taxonomy of finite automata construction algorithms. Technical Report Computing Science Note 93/43, Eindhoven University of Technology, 1994.