

THE SOLUTIONS OF TWO STAR-HEIGHT PROBLEMS FOR REGULAR TREES

J.P. BRAQUELAIRE and B. COURCELLE

Department of Mathematics and Computer Science, Bordeaux-I University, 33405 Talence, France

Communicated by M. Nivat

Received May 1983

Revised August 1983

Abstract. Regular trees can be defined by two types of rational expressions. For these two types we solve the star-height problem, i.e., we show how to construct a rational expression of minimal star-height from the minimal graph of the given tree (i.e., the analogue of the minimal deterministic automaton for regular languages). In one case, the minimal starheight is the rank (in the sense of Eggan) of the minimal graph. There corresponds a characterization of the star-height of a prefix-free regular language w.r.t. rational expressions of a special kind (called deterministic) as the rank of its minimal deterministic automaton considered as a graph.

Contents

Introduction	205
1. Preliminaries. Regular trees and rational expressions	207
2. Constructing a rational expression from a ranked graph	213
3. The star-height of a regular tree	217
4. Application to other types of rational expressions	219
5. The restricted star-height of regular trees	225
Appendix A	233
References	239

Introduction

Regular trees, i.e., trees which are either finite or infinite with only finitely many distinct subtrees, play an important role in the theory of program schemes. They have been investigated by Cousineau [9], Jacob [16], Elgot et al. [13] and Courcelle [8].

Since they form the free iterative theory (generated by some ranked alphabet F), they are denoted by certain *iterative theory expressions* (see [2, 13, 14]). These iterative theory expressions include the *rational expressions* independently defined by Cousineau [9]. The relation between these two classes of expressions has been shown by Courcelle [8].

All these expressions use an iteration operator (denoted \dagger or $*$) very close to Kleene's $*$ for languages. They raise a *star-height problem*, i.e., the problem of constructing a rational expression of minimal star-height which defines a given regular tree.

This problem is trivial for iterative theory expressions which use vector iteration since every regular tree can be defined by such an expression with one iteration if the tree is infinite and no iteration if it is finite [12]. It is not if iterative theory expressions are restricted so as to use only *scalar iteration*. We solve it and we show that the minimal star-height is exactly the rank of the minimal graph of the tree (the rank of a directed graph has been introduced by Eggan [10] for the study of rational expressions defining languages and further investigated by McNaughton [17, 18] and Cohen and Brzozowski [3–6]).

These expressions use an operation called *composition*, a typical case of which is $e_{\sigma}(e_1, \dots, e_n)$, which denotes the tree obtained by the substitution of $\mathbf{Val}(e_1), \dots, \mathbf{Val}(e_n)$ at certain leaves of $\mathbf{Val}(e)$ (we denote by $\mathbf{Val}(e)$ the tree defined by the expression e).

The major contribution of Cousineau was to show that the operation of composition is dispensable and that the resulting expressions still generate all regular trees (see [8] for a simple proof). These restricted expressions raise another star-height problem for which we also give the solution. The minimal star-height in this sense is also obtained from the consideration of the minimal graph of the tree.

For technical reasons, we shall work neither with iterative theory expressions [2, 8, 13] nor with rational expressions [8, 9] but with slightly different expressions (still called *rational*) which use the following constructions:

$*_v(e)$: iterate $\mathbf{Val}(e)$ with respect to the variable v .

$e_{\sigma(v_1, \dots, v_k)}(e_1, \dots, e_k)$: substitute $\mathbf{Val}(e_1), \dots, \mathbf{Val}(e_k)$ for v_1, \dots, v_k in $\mathbf{Val}(e)$.

Our results will be obtained for these rational expressions but they transfer easily to the above mentioned expressions.

The proofs of our two results follow the same pattern that can be sketched as follows.

A regular tree is manipulated by means of a finite pointed graph of which it is the infinite unlooping. These graphs can be 'structured', in different ways, but each 'structuring' is characterized by an integer, its 'depth'.

For each structuring of 'depth' n , one can construct a rational expression of star-height n . Hence, a certain rational expression can be associated with a 'structuring' of minimal 'depth' of the minimal pointed graph of the given tree.

It turns out that this rational expression is the right one, i.e., is of minimal star-height among all those defining the given tree.

In order to prove this, we first define some syntactical manipulations performing some simplifications of rational expressions. They transform a rational expression into an equivalent one in *normal form*.

From the syntactical structuring of a minimal rational expression in normal form, one can construct a 'structuring' of the minimal graph of the tree whose 'depth' is not less than the star-height of the expression. And from the first construction one gets an equality as required.

1. Preliminaries. Regular trees and rational expressions

We recall quickly from [8] the following terminology and notations:

- F is a finite ranked alphabet,
- $\rho(f)$ for $f \in F$ is the rank (≥ 0) of the function symbol f ,
- $F_k = \{f \in F \mid \rho(f) = k\}$,
- V is an infinite set of variables (of arity 0),
- $M(F, V)$ is the set of finite well-formed terms built on $F \cup V$ and identified in a classical way with finite trees,
- $M^\infty(F, V)$ is the set of finite and infinite such trees,
- $R(F, V)$ is the set of finite and infinite regular trees, i.e., of trees T in $M^\infty(F, V)$ such that the set **Subtree**(T) of their subtrees is finite,
- $\mathbf{Var}(T)$ for T in $M^\infty(F, V)$ is the set of variables having an occurrence in T .

1.1. Definition. $E(F, V)$ is the set of *rational expressions* (r.e.) we shall use in this paper; it is inductively defined as follows:

- (i) v in V is a r.e.
- (ii) a in F_0 is a r.e.
- (iii) $f(e_1, \dots, e_k)$ is a r.e. if $f \in F_k$ and e_1, \dots, e_k are r.e.
- (iv) $*_v(e')$ is a r.e. if $v \in V$ and e' is a r.e.
- (v) $e_{0,\sigma}(e_1, \dots, e_k)$ is a r.e. if e_0, e_1, \dots, e_k are r.e. and σ is a sequence (v_1, \dots, v_k) of distinct variables of V .

We shall denote by $\sigma(i)$ the i th element of σ ($1 \leq i \leq k$). We shall also write e', e'' instead of $e'_{(v)}$, e'' . We shall denote by $E_0(F, V)$ the set of *restricted rational expressions* (r.r.e), i.e., the subset of $E(F, V)$ consisting of r.e. built without rule (v).

For trees T, T_1, \dots, T_k we denote by $T[T_1/v_1, \dots, T_k/v_k]$ the substitution of T_i for v_i in T , where v_1, \dots, v_k are distinct variables in V .

1.2. Definition. The value $\mathbf{Val}(e)$ of a rational expression e is a regular tree defined as follows:

$$\mathbf{Val}(e) = \begin{cases} a & \text{if } e = a, a \in F_0, \\ v & \text{if } e = v, v \in V, \\ f(\mathbf{Val}(e_1), \dots, \mathbf{Val}(e_k)) & \text{if } e = f(e_1, \dots, e_k), \\ \mathbf{Val}(e_0)[\mathbf{Val}(e_1)/\sigma(1), \dots, \mathbf{Val}(e_k)/\sigma(k)] & \text{if } e = e_{0,\sigma}(e_1, \dots, e_k), \\ \text{the unique tree } t \text{ such that } t = \mathbf{Val}(e')[t/v] & \text{if } e = *_v(e') \text{ and } \mathbf{Val}(e') \neq v. \end{cases}$$

The tree $\mathbf{Val}(e)$ is undefined if in the last case $\mathbf{Val}(e') = v$ or $\mathbf{Val}(e')$ or any of the required $\mathbf{Val}(e_0), \dots, \mathbf{Val}(e_k)$ is undefined (depending on the case).

It is easy to decide whether $\mathbf{Val}(e)$ is defined or not.

Convention. We shall only consider r.e.'s e such that $\mathbf{Val}(e)$ is defined.

Two r.e.'s are *equivalent* if they define the same tree.

A few more definitions concerning the syntax of r.e. will be necessary.

1.3. Definition. The set $\mathbf{Sub}(e)$ of the *subexpressions* of an r.e. e defined as in Definition 1.1 is given by

$$\begin{aligned} \{e\} & \text{ in cases (i) and (ii),} \\ \{e\} \cup \mathbf{Sub}(e_1) \cup \dots \cup \mathbf{Sub}(e_k) & \text{ in case (iii),} \\ \{e\} \cup \mathbf{Sub}(e') & \text{ in case (iv),} \\ \{e\} \cup \mathbf{Sub}(e_0) \cup \dots \cup \mathbf{Sub}(e_k) & \text{ in case (v).} \end{aligned}$$

From Definition 1.2 it is easy to see that if a subexpression e' of e is replaced by an equivalent r.e. \bar{e}' at some of its occurrences in e , then the resulting r.e. \bar{e} is equivalent to e .

1.4. Definition. All occurrences of the variable v in $*_i(e')$ are *bound*; if $\sigma(i) = v$ for some i , the occurrences of v in e' are *bound* in $e'_{\cdot, \sigma}(e''_1, \dots, e''_k)$.

An occurrence of v which is not bound is *free*. Bound variables can be renamed provided classical conflicts are avoided. Note that v may be both free and bound in $e'_{\cdot, \sigma}(e''_1, \dots, e''_k)$ (if $\sigma(i) = v$ and v is free in e''_1, \dots, e''_k). As in the lambda-calculus for instance in [1], we shall consider as identical two r.e. e and e' which only differ by renaming of bound variables and we shall write $e = e'$.

Finally, we denote by $\mathbf{Fvar}(e)$ the set of variables having free occurrences in e . It is clear that $\mathbf{Fvar}(e) \supseteq \mathbf{Var}(\mathbf{Val}(e))$ but the inclusion may be strict. Take for example $e = (v_1)_{\cdot, (v_1, v_2)}(v_3, v_4)$. Then $\mathbf{Fvar}(e) = \{v_3, v_4\}$ and $\mathbf{Val}(e) = v_3$ so that $\mathbf{Var}(\mathbf{Val}(e)) \neq \mathbf{Fvar}(e)$.

1.5. Definition. The *star-height* of e in $E(F, V)$ is defined as follows:

$$h(e) = \begin{cases} 0 & \text{if } e \in F_0 \cup V, \\ \mathbf{Max}\{h(e_1), \dots, h(e_k)\} & \text{if } e = f(e_1, \dots, e_k) \text{ or } e = e_{1, \sigma}(e_2, \dots, e_k), \\ 1 + h(e') & \text{if } e = *_i(e'). \end{cases}$$

The *star-height* of a regular tree T is the minimum star-height of an r.e. with value T , and will be denoted by $h(T)$. An r.e. e is of *minimal star-height* if $h(e) = h(\mathbf{Val}(e))$. The *restricted star-height* of a regular tree T is the minimum star-height of an r.r.e. with value T . We denote it by $h_0(T)$.

When saying that an r.r.e. is of *minimal star-height* we shall mean that it has this property among the r.r.e.'s defining the same tree. We shall also use the *length* of a r.e., denoted by $|e|$ and defined as follows:

$$|e| = 1 \quad \text{if } e \in F_0 \cup V,$$

$$|f(e_1, \dots, e_k)| = 1 + |e_1| + \dots + |e_k|,$$

$$|*_t(e)| = 1 + |e|,$$

$$|e_{0,\sigma}(e_1, \dots, e_k)| = |e_0| + |e_1| + \dots + |e_k| \quad \text{where } k \neq 0.$$

It follows that $|e| < |e'|$ when e is a sub-expression of e' which is proper, i.e., $e \neq e'$.

Finally, the notation $e_{0,\sigma}(e_1, \dots, e_k)$ will be (exceptionally) extended to the case $k = 0$ (i.e., $\sigma = \varepsilon$) and it means e_0 .

Our constructions will make a heavy use of various types of graphs.

1.6. Definition. By a *graph* we shall mean a pair $G = (N, A)$, consisting of a finite set of *nodes* N and a finite set of *arcs* A . Each arc a in A has a *source* and a *target* in N .

A *subgraph* of G is a graph $G' = (N', A')$ such that $N' \subseteq N$ and $A' \subseteq A$. It is *full* if every arc of A having a source and a target in N' is in A' . If $N'' \subseteq N$ we denote by $G \setminus N''$ the full subgraph of G with set of nodes $N - N''$. It may be empty if $N'' = N$.

If one has a *path* a_1, a_2, \dots, a_n , $n \geq 1$, from s to s' (i.e., s is the source of a_1 , s' the target of a_n , the source of a_{i+1} is the target of a_i), then s' is a *descendant* of s and s is an *ancestor* of s' . A path from s to s is a *cycle*. We denote by $G \downarrow s$ the full subgraph of G whose nodes are s and its descendants in G .

A *pointed graph* is a pair (G, s) of a graph and one of its nodes.

A homomorphism $\varphi: G \rightarrow G'$ is a pair of mappings: $N \rightarrow N'$ and $A \rightarrow A'$ both denoted by φ , such that sources and targets are preserved in an obvious way.

1.7. Definition. Given F and V , we define a *ranked graph* as a graph satisfying the following extra conditions:

- (i) each node s has a *label* in $F \cup V$ denoted by $\mathbf{Lab}(s)$,
- (ii) for each node s , an order is given on the set of arcs with source s ,
- (iii) for each node s , the number of arcs with source s is equal to $\rho(\mathbf{Lab}(s))$.

It is convenient to define a ranked graph as a triple $G = (N, \mathbf{Suc}, \mathbf{Lab})$, where N is the set of nodes. $\mathbf{Lab}(n)$ is the label of node n , \mathbf{Suc} is a mapping: $N \rightarrow N^*$ defining the sequence of *successors* of a node, i.e., the sequence of targets of arcs (in the order defined by (ii)) having that node for source

Hence, every ranked graph has an underlying graph. All definitions given w.r.t. graphs will apply to a ranked graph by the intermediate of its underlying graph. Note that a subgraph of a ranked graph is not necessarily the underlying graph of a ranked graph.

A homomorphism $\varphi: G \rightarrow G'$ is a mapping $\varphi: N \rightarrow N'$ such that $\mathbf{Lab} = \mathbf{Lab}' \circ \varphi$ and $\varphi \circ \mathbf{Suc} = \mathbf{Suc}' \circ \varphi$ (labels and successors are preserved).

1.8. Definition (using terminology and results of Courcelle [8]). A regular tree can be associated as follows with a pointed ranked graph (G, s) , where $G = (N, \mathbf{Suc}, \mathbf{Lab})$. First we associate with G a regular system of equations $\Sigma_G = \langle w_n = t_n; n \in N \rangle$ with set of unknowns $W = \{w_n \mid n \in N\}$ and t_n being defined as follows:

$$t_n = \begin{cases} a & \text{if } \mathbf{Lab}(n) = a \in F_0, \\ v & \text{if } \mathbf{Lab}(n) = v \in V, \\ f(w_{n_1}, \dots, w_{n_k}) & \text{if } f \in F_k, k \geq 1, \mathbf{Suc}(n) = (n_1, \dots, n_k). \end{cases}$$

This system has a unique solution in $M^\infty(F, V)$ consisting of a family $\langle T_n \mid n \in N \rangle$ of regular trees. We associate T_s with (G, s) and denote it by $T(G, s)$.

Let now T be a given regular tree. There exists a canonical pointed ranked graph (G, s) such that $T(G, s) = T$, denoted by (G_T, n_T) and constructed as follows:

$$\begin{aligned} G_T &= (\mathbf{Subtree}(T), \mathbf{Suc}, \mathbf{Lab}), & n_T &= T, \\ \mathbf{Suc}(U) &= (U_1, \dots, U_k), \\ \mathbf{Lab}(U) &= f \text{ whenever } U = f(U_1, \dots, U_k), f \in F_k, \\ & & & U_1, \dots, U_k \in \mathbf{Subtree}(T). \end{aligned}$$

1.9. Proposition. (1) (G_T, n_T) is the unique (up to isomorphism) pointed ranked graph (G, s) with minimal number of nodes and such that $T(G, s) = T$.

(2) For every pointed ranked graph (G, s) such that $G = G \upharpoonright s$ and $T = T(G, s)$, there exists a unique homomorphism $\varphi: G \rightarrow G_T$ such that $\varphi(s) = n_T$. This homomorphism is surjective.

We call (G_T, n_T) the minimal pointed graph of T and G_T its minimal graph (omitting the word 'ranked'). This graph corresponds to the minimal deterministic automaton recognizing a regular language. The proof of Proposition 1.9 is fully similar to that of the corresponding proposition for regular languages (see [11]).

1.10. Definition. A graph is *strongly connected* (s.c.) if for every pair of distinct nodes (n, n') there is a path from n to n' . A *strongly connected component* (s.c.c.) of a graph G is a maximal s.c. subgraph of G , i.e., the full subgraph of G with set of nodes $N_n = \{n' \mid n' \in G \upharpoonright n \text{ and } n \in G \upharpoonright n'\}$ for some n .

Such a s.c.c. is *trivial* if it is reduced to a single node with no arc. (If it is reduced to a single node n together with loops, i.e., arcs from n to n , it is not trivial.)

We denote by $\mathbf{SCC}(G)$ the set of nontrivial s.c.c.'s of a graph G .

1.11. Definition. We can now define the *rank* of a graph G with set of nodes N as follows:

- (0) if G is empty or $\text{SCC}(G) = \emptyset$, then $r(G) = 0$,
 (1) if G is not empty and $\text{SCC}(G) = \{G\}$, then $r(G) = 1 + \text{Min}\{r(G \setminus n) \mid n \in N\}$,
 (2) otherwise, $r(G) = \text{Max}\{r(G') \mid G' \in \text{SCC}(G)\}$.

Note that $r(G) = 1$ if G consists of one node with one loop and that $r(G) = 0$ if G has no cycle. This concept of rank has been introduced by Eggan [10] who showed that, for every regular language R defined by a transition graph G , a rational expression of star-height $r(G)$ could be constructed to define it, and conversely that a transition graph of rank $h(e)$ could be associated with a rational expression e . Cohen [5] has shown that the star-height of R is exactly the minimal rank of the graph corresponding to a non-deterministic finite automaton without ε -transitions recognizing R (but this does not give any algorithm to compute the star-height of R). But for certain types of finite automata (called reset-free, with single final-state), the star-height of R is the rank of the minimal deterministic automaton [3, 4].

We can state a similar result.

1.12. Theorem. *The star-height of a regular tree is the rank of its minimal graph.*

The restricted star-height of a regular tree will be characterized in terms of another concept, the *length* of a pointed graph that we now define.

1.13. Definition. Let G be a graph with set of nodes N .

A subset B of N is called a *base* of G if every cycle of G has at least one node in B . A base is *minimal* if no proper subset of B is a base. A *B -simple* path is a path where no node of B occurs more than once. By $B \cap \pi$ we denote the set of nodes of B appearing in a path π .

We now define the *length* $l(G, n)$ of a pointed graph (G, n) as the integer

$$l(G, n) = \text{Min}\{l_B(G, n) \mid B \text{ is a base of } G\},$$

where

$$l_B(G, n) = \text{Max}\{\text{Card}(B \cap \pi) \mid \pi \text{ is a } B\text{-simple path starting at } n\}.$$

Remark that **Min** can be restricted to the minimal bases of G . If G has no cycle, then \emptyset is a base and $l(G, n) = 0$.

1.14 Theorem. *The restricted star-height of a regular tree is the length of its minimal pointed graph.*

We shall close this section by defining some operations on graphs.

1.15. Definition. Let (G_i, n_i) , $i = 0, \dots, k$, be pointed ranked graphs, and $T_i = T(G_i, n_i)$ for all i . Let v_1, \dots, v_k be distinct variables in V . Let us also assume that $G_i = (N_i, \text{Suc}_i, \text{Lab}_i)$ with $N_i \cap N_j = \emptyset$ for $i \neq j$.

We denote by $(G_0, n_0)[(G_1, n_1)/v_1, \dots, (G_k, n_k)/v_k]$ the pointed ranked graph (G, n) with $G = (N, \mathbf{Suc}, \mathbf{Lab})$ defined as follows:

$$N = N'_0 \cup N_1 \cup \dots \cup N_k,$$

$$N'_0 = N_0 - \{s \in N_0 \mid \mathbf{Lab}_0(s) \in \{v_1, \dots, v_k\}\},$$

$$\mathbf{Suc}(s) = \begin{cases} \mathbf{Suc}_i(s) & \text{if } s \in N_i, 1 \leq i \leq k, \\ (s'_1, \dots, s'_l) & \text{if } s \in N'_0, \mathbf{Suc}_0(s) = (s_1, \dots, s_l) \text{ and} \\ & s'_j = \text{if } \mathbf{Lab}_0(s_j) = v_i \text{ and } 1 \leq i \leq k \text{ then } n_i \text{ else } s_j, \end{cases}$$

$$\mathbf{Lab}(s) = \mathbf{Lab}_i(s) \quad \text{if } s \in N_i, 0 \leq i \leq k,$$

$$n = \text{if } \mathbf{Lab}(n_0) = v_i \text{ and } 1 \leq i \leq k \text{ then } n_i \text{ else } n_0.$$

Remark that if $G_i \upharpoonright n_i = G_i$ for all $i = 0, 1, \dots, k$ and each v_i labels at least one node of G_0 , then $G \upharpoonright n = G$.

1.16. Proposition. *With the above notations,*

$$T(G, n) = T(G_0, n_0)[T(G_1, n_1)/v_1, \dots, T(G_k, n_k)/v_k].$$

Proof. The proof follows from the consideration of the systems $\Sigma_{G_0}, \dots, \Sigma_{G_k}$ and Σ_G . \square

1.17. Definition. Let (G_i, n_i) , $i = 1, \dots, k$, be as in Definition 1.15 and let $f \in F_k$, $k \geq 1$.

We denote by $f((G_1, n_1), \dots, (G_k, n_k))$ the pointed ranked graph (G, n) with $G = (N, \mathbf{Suc}, \mathbf{Lab})$, and where

$$N = N_1 \cup \dots \cup N_k \cup \{n\}, \quad n \notin N_1, \dots, N_k,$$

$$\mathbf{Suc}(n) = (n_1, \dots, n_k),$$

$$\mathbf{Suc}(s) = \mathbf{Suc}_i(s) \quad \text{for } s \text{ in } N_i, 1 \leq i \leq k,$$

$$\mathbf{Lab}(n) = f,$$

$$\mathbf{Lab}(s) = \mathbf{Lab}_i(s) \quad \text{for } s \text{ in } N_i, 1 \leq i \leq k.$$

1.18. Proposition. $T(G, n) = f(T(G_1, n_1), \dots, T(G_k, n_k))$.

1.19. Definition. Let $G = (N, \mathbf{Suc}, \mathbf{Lab})$ be a ranked graph, v be a variable, n one of its nodes not labeled by v and such that v labels at least one descendant of n . We denote by $*_v(G, n)$ the pointed ranked graph (G', n) where $G' = (N', \mathbf{Suc}', \mathbf{Lab}')$ such that

$$N' = N - \{s \in N \mid \mathbf{Lab}(s) = v\},$$

$$\mathbf{Lab}'(s) = \mathbf{Lab}(s) \quad \text{for } s \in N',$$

$$\mathbf{Suc}'(s) = (s'_1, \dots, s'_k), \quad \text{where } \mathbf{Suc}(s) = (s_1, \dots, s_k) \text{ and}$$

$$s'_i = \text{if } \mathbf{Lab}(s_i) = v \text{ then } n \text{ else } s_i.$$

Intuitively, G' is obtained from G by identifying with n all nodes labelled by v .

1.20. Proposition. $T(*_v(G, n)) = *_v(T(G, n))$.

We close this preliminary section with two more lemmas concerning rational expressions.

1.21. Lemma. *If $v \notin \sigma$ and is not free in (e_1, \dots, e_k) , then $*_v(e_{\cdot\sigma}(e_1, \dots, e_k))$ is equivalent to $*_v(e)_{\cdot\sigma}(e_1, \dots, e_k)$.*

Proof. It suffices to show that if $T \in M^{\omega}(F, V)$, $\mathbf{Val}(e) \neq v$, $\mathbf{Var}(T) \cap \{\sigma(1), \dots, \sigma(k), v\} = \emptyset$ and $T = \mathbf{Val}(e)[T/v]$, then

$$\begin{aligned} T[\mathbf{Val}(e_1)/\sigma(1), \dots, \mathbf{Val}(e_k)/\sigma(k)] &= \\ &= \mathbf{Val}(e)[T[\mathbf{Val}(e_1)/\sigma(1), \dots, \mathbf{Val}(e_k)/\sigma(k)]/v], \end{aligned}$$

and this follows from [8, Proposition 3.4.2]. \square

Remark that this lemma does not hold for the r.e. introduced by Cousineau (see their formal definition in Sections 4 and 5 below).

Notation. For e in $E(F, V)$ and $v_1, v_2, \dots, v_k, w_1, \dots, w_k$ in V , we denote by $e[v_1/w_1, \dots, v_k/w_k]$ the result of the substitution in e of v_i for all *free* occurrences of w_i , $1 \leq i \leq k$. We denote by \equiv the equivalence of r.e.'s.

1.22. Lemma. *We have $*_v(*_w(e)) \equiv *_w(*_v(e)) \equiv *_v(e[v/w])$.*

Proof. The proof follows by similar computations as above. \square

2. Constructing a rational expression from a ranked graph

The first half of the proof of our main theorem consists in establishing the following analogue of Eggen's theorem.

2.1. Theorem. *For every regular tree T , one can construct a rational expression defining Γ with star-height at most $r(G_T)$.*

We first introduce a set of notations, remarks and technical results.

2.2. Notations. For graphs G and G' ,

- $|G|$ is the number of nodes of G ,
- $G \approx G'$ means that G and G' are isomorphic; by $G = G'$ we shall mean that G and G' have same nodes and same arcs,
- $\mathbf{Core}(G)$ for a ranked graph G denotes the (non-ranked) graph $G \setminus N'$ where N' is the set of its nodes labeled in $F_0 \cup V$,
- $\|G\| = |\mathbf{Core}(G)|$.

2.3. Remark. For any ranked graph G ,

$$\text{SCC}(\text{Core}(G)) = \text{SCC}(G). \quad r(\text{Core}(G)) = r(G).$$

2.4. Proposition [5, Proposition 2.2]. *If G is a subgraph of G' , then $r(G) \leq r(G')$.*

2.5. Proposition. *Let (G, n) be a pointed ranked graph such that $G \upharpoonright n = G$ and let $T = T(G, n)$. Then $r(G_T) \leq r(G)$.*

Proof. Recall from Proposition 1.9 that there exists a unique surjective homomorphism $\varphi: G \rightarrow G_T$ such that $\varphi(n) = n_T$. In fact, $\varphi(s) = T(G, s)$ (which is a node of G_T by the construction we gave of G_T) for all modes s of G .

A path in a ranked graph like G can be represented as a sequence $\pi = (n_j, i_1, n_2, i_2, \dots, i_{m-1}, n_m)$, where n_1, n_2, \dots, n_m are nodes and i_1, \dots, i_{m-1} are integers such that n_{j+1} is the i_j th successor of n_j . Its image under φ , namely, $\varphi(\pi) = (\varphi(n_1), i_1, \dots, i_{m-1}, \varphi(n_m))$, is a path in G_T .

Let us show that, conversely, every path π' of G_T is $\varphi(\pi)$ for some path π in G . This will show that φ is a pathwise homomorphism in the sense of McNaughton [17] (see also [5]), hence, by a theorem of [17], that $r(G_T) \leq r(G)$.

Let $\pi' = (t_1, i_1, t_2, i_2, \dots, t_m)$. Let n_1 be any node of G such that $\varphi(n_1) = t_1$ (there exists some since φ is surjective). Then define n_2 as the i_1 th successor of n_1 . This is possible since n_1 and t_2 have the same label, say f . Note that $t_1 = f(t'_1, \dots, t'_k)$ for some f and $t_2 = t'_i$. Hence $t_2 = T(G, n_2) = \varphi(n_2)$.

We can go on and define n_3 from n_2 and i_2 , and prove similarly that $\varphi(n_3) = t_3$. By repeating the construction one gets a path $\pi = (n_1, i_1, n_2, \dots, n_m)$ and the proof that $\varphi(\pi) = \pi'$. \square

Theorem 2.1 will be a corollary of the following more general proposition.

2.6. Proposition. *Let G be a ranked graph. For each node n of G , one can construct a rational expression e such that*

- (1) $h(e) \leq r(G)$,
- (2) $\text{Val}(e) = T(G, n)$.

Proof. We do the proof by induction on $r(G)$ with $G = (N, \text{Suc}, \text{Lab})$.

Basis. If $r(G) = 0$, this means that G has no cycle, hence that $T(G, n)$ is finite; one can simply take $e = T(G, n)$, or that G is empty and the result holds trivially.

Inductive step: Let us now consider the case $r(G) > 0$. Assuming the result for all graphs of lower rank, we do the proof by using an induction on $\|G\| = |\text{Core}(G)|$.

Case 1. $\text{Core}(G)$ is not strongly connected.

Given a node n we let H be its s.c.c. in $\text{Core}(G)$. It is clear that $|H| < |\text{Core}(G)| = \|G\|$.

Let n_1, \dots, n_k be the nodes of G which do not belong to H but are successors (w.r.t. G) of nodes in H .

Let v_1, \dots, v_k be variables not occurring in G .

Let $G_0 = (N_0, \mathbf{Suc}_0, \mathbf{Lab}_0)$ be the ranked graph such that

$$N_0 = \{n_1, \dots, n_k\} \cup \{\text{the nodes of } H\},$$

$$\mathbf{Suc}_0(s) = \begin{cases} \mathbf{Suc}(s) & \text{for all } s \text{ in } H, \\ \varepsilon & \text{for } s \text{ in } \{n_1, \dots, n_k\}, \end{cases}$$

$$\mathbf{Lab}_0(s) = \begin{cases} \mathbf{Lab}(s) & \text{for all } s \text{ in } H, \\ v_i & \text{for } s = n_i, 1 \leq i \leq k. \end{cases}$$

Note that $\mathbf{Core}(G_0) = H$ whence $\|G_0\| < \|G\|$. Also $r(G_0) \leq r(G)$ since G_0 is a subgraph of G and by Proposition 2.4. Hence, by induction, we can assume that we have a r.e. e_0 of star-height at most $r(G_0)$ defining $T(G_0, n)$.

Note that

$$T(G, n) = T(G_0, n)[T_1/v_1, \dots, T_k/v_k], \quad \text{with } T_i = T(G, n_i) \text{ for } i = 1, \dots, k.$$

For all i , $1 \leq i \leq k$, we have $T(G, n_i) = T(G_i, n_i)$ where $G_i = G \upharpoonright n_i$ and $\mathbf{Core}(G_i)$ is a proper subgraph of $\mathbf{Core}(G)$ since it does not contain at least the node n .

Note also that $r(G_i) = \mathbf{Max}\{r(G') \mid G' \in \mathbf{SCC}(\mathbf{Core}(G_i))\}$ by Remark 2.3. Since $H \in \mathbf{SCC}(\mathbf{Core}(G))$ and $\mathbf{SCC}(\mathbf{Core}(G_i)) \subseteq \mathbf{SCC}(\mathbf{Core}(G))$ we have

$$\begin{aligned} r(G) &\geq \mathbf{Max}\{r(H), r(\mathbf{Core}(G_1)), \dots, r(\mathbf{Core}(G_k))\} \\ &\geq \mathbf{Max}\{r(G_0), r(G_1), \dots, r(G_k)\}. \end{aligned}$$

In particular, for all $i = 1, \dots, k$,

$$r(G_i) \leq r(G), \quad \|G_i\| < \|G\|,$$

so that by induction we can assume the existence of e_1, \dots, e_k such that

$$h(e_i) \leq r(G_i), \quad \mathbf{Val}(e_i) = T_i.$$

Hence the r.e. $e = e_{0,\sigma}(e_1, \dots, e_k)$ with $\sigma = (v_1, \dots, v_k)$ is such that

$$\mathbf{Val}(e) = T, \quad h(e) = \mathbf{Max}\{h(e_0), h(e_1), \dots, h(e_k)\} \leq r(G).$$

Case 2. $\mathbf{Core}(G)$ is strongly connected.

Let p be a cycle center [5], i.e. a node in $\mathbf{Core}(G)$ such that

$$r(\mathbf{Core}(G) \setminus p) = r(\mathbf{Core}(G)) - 1 = r(G) - 1.$$

Necessarily, $\mathbf{Lab}(p)$ is some symbol f of arity $k \geq 1$. Let G_0 be the ranked graph $(N, \mathbf{Suc}_0, \mathbf{Lab}_0)$ with

$$\mathbf{Suc}_0(s) = \mathbf{Suc}(s) \quad \text{if } s \neq p,$$

$$\mathbf{Suc}_0(p) = \varepsilon,$$

$$\mathbf{Lab}_0(s) = \mathbf{Lab}(s) \quad \text{if } s \neq p,$$

$$\mathbf{Lab}_0(p) = v, \quad \text{where } v \text{ is a variable not appearing in } G.$$

It is clear that $\mathbf{Core}(G_0) = \mathbf{Core}(G) \setminus p$ and that $r(G_0) = r(G) - 1$. Let also (n_1, \dots, n_k) be $\mathbf{Suc}(p)$. Some of the nodes n_1, \dots, n_k may be equal to p . Without loss of generality we shall assume that $\mathbf{Suc}(p) = (n_1, \dots, n_l, p, p, \dots, p)$ with $n_i \neq p$ for $i = 1, \dots, l$.

Hence, by induction, there exist r.e.'s e_1, \dots, e_l with

$$\mathbf{Val}(e_i) = T(G_0, n_i), \quad h(e_i) \leq r(G_0) = r(G) - 1$$

for $i = 1, \dots, l$. Note that

$$\begin{aligned} T(G, n_i) &= T(G_0, n_i)[T(G, p)/v], \\ T(G, p) &= f(T(G, n_1), \dots, T(G, n_l), T(G, p), \dots, T(G, p)) \\ &= \mathbf{Val}(e')[T(G, p)/v], \end{aligned}$$

where e' is the r.e. (of star-height at most $r(G) - 1$) $f(e_1, \dots, e_l, v, v, \dots, v)$.

It follows that

$$T(G, p) = \mathbf{Val}(*_v(e')), \quad h(*_v(e')) \leq r(G).$$

Going back to the initial problem, i.e., to find e defining $T(G, n)$ we have two cases. If $n = p$ then we are done. Otherwise, as for the n_i 's we have

$$T(G, n) = T(G_0, n)[T(G, p)/v].$$

Assuming by induction that we already have e_0 such that

$$\mathbf{Val}(e_0) = T(G_0, n), \quad h(e_0) \leq r(G) - 1,$$

we can take $e = e_{0,r}(*_r(e'))$ of star-height at most $r(G)$ to define $T(G, n)$. \square

Proof of Theorem 2.1. By applying Proposition 2.6 to (G_T, n_T) one gets a rational expression defining $T(G_T, n_T) = T$ of star-height at most $r(G_T)$. \square

2.7. Remark. The construction of Proposition 2.6 is effective. By brute force enumeration it is always possible to find the necessary p for Case 2. The rank of a graph can be determined similarly. We shall not consider here the problem of finding efficient algorithms for doing this.

2.8. Example. Let Σ be the following regular system:

$$x = f(x, y), \quad y = g(y, x, z), \quad z = g(z, y, t), \quad t = h(t, t),$$

to which correspond the 4-tuple (X, Y, Z, T) of regular trees as unique solution.

The trees X, Y, Z have the same minimal graph G depicted in Fig. 1(a). Its nodes are denoted by x, y, z, t and are labeled respectively by $f, g, g,$ and h . The rank of G is 2 with cycle center y .

Let us construct a r.e. of star-height 2 which defines X .

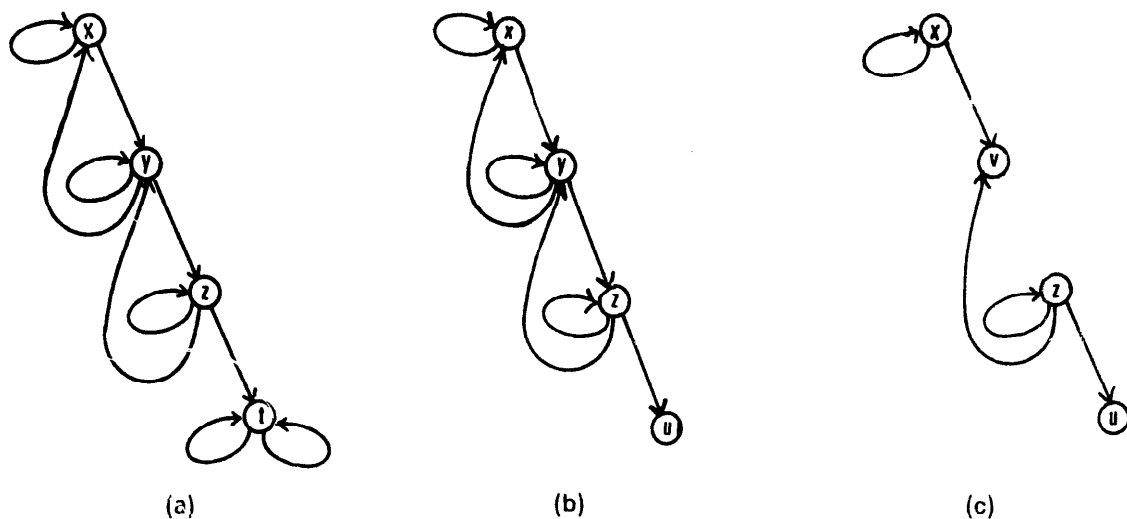


Fig. 1.

The graph G has two s.c. components: the full subgraphs with sets of nodes $\{x, y, z\}$ and $\{t\}$. The second one corresponds to the tree T and one can define it by $e_0 = *_v(h(v, v))$.

Hence we shall define X by $e_{1..u}e_0$, where e_1 is obtained from the graph G' of Fig. 1(b) and is assumed to define $T(G', x)$. Since y is a cycle center of G' , we consider G'' shown in Fig. 1(c), which is exactly $G' \setminus y$ completed with some arcs with target v so as to get a ranked graph (i.e., the graph G_0 of the proof of Proposition 2.6, Case 2).

One gets $e_2 = *_w(f(w, v))$ for defining $T(G'', x)$, and $e_3 = *_w(g(w, v, u))$ for defining $T(G'', z)$. The tree $T(G', y)$ can now be defined by $e_4 = *_v(g(v, e_2, e_3))$.

Recall that we want e_1 defining $T(G', x)$ (remark that e_2 was 'only' defining $T(G'', x)$). We can take $e_1 = e_{2..v}e_4$ and finally $e = (e_{2..v}e_1)_{..u}e_0$ to define $X = T(G, x)$ as wanted.

3. The star-height of a regular tree

We prove here that the construction used for the proof of the Theorem 2.1 is optimal w.r.t. star-height.

We shall need a certain notion of normal form for rational expressions.

3.1. Definition. Let M be the set of rational expressions e satisfying the following conditions:

- (1) e and all its subexpressions are of minimal star-height,
- (2) if $e_{0..v}(e_1, \dots, e_k)$ is a subexpression of e , then each $\sigma(i)$ belongs to $\mathbf{Var}(\mathbf{Val}(e_0))$,
- (3) if $*_v(e_0)$ is a subexpression of e , then e_0 is of the form $f(e_1, \dots, e_k)$ and $v \in \mathbf{Var}(\mathbf{Val}(e_0))$.

3.2. Proposition. *Every rational expression is equivalent to some rational expression in M .*

The proof will be given in Appendix A.

3.3. Theorem. *Let e be an r.e. in M , let $T = \mathbf{Val}(e)$ and G_T be the minimal graph of T . Then $h(e) = r(G_T)$.*

Proof. Since every subexpression of e is in M , it will be possible to do the proof by induction on the structure of e .

Case 1. $e \in F_0 \cup V$. Then $h(e) = r(G_T) = 0$.

Case 2. $e = f(e_1, \dots, e_k)$.

Let $T = \mathbf{Val}(e)$, $T_i = \mathbf{Val}(e_i)$ so that $T = f(T_1, \dots, T_k)$. Since $h(e) = \mathbf{Max}\{h(e_1), \dots, h(e_k)\}$ and, by induction, $h(e_i) = r(G_{T_i})$ for all i we need only show that $r(G_T) = \mathbf{Max}\{r(G_{T_1}), \dots, r(G_{T_k})\}$.

Note that $G_{T_i} \approx G_T \upharpoonright n_i$, where (n_1, \dots, n_k) is the list of successors of n_T in G_T .

Subcase 1. n_T is not the target of any arc in G_T . Then clearly $\mathbf{SCC}(G_T) = \mathbf{SCC}(G_T \setminus n_T) \approx \mathbf{SCC}(G_{T_1}) \cup \dots \cup \mathbf{SCC}(G_{T_k})$ and the result follows. (For two sets of graphs A and B , $A \approx B$ means that for all graph H in A there exists H' in B such that $H \approx H'$ and vice-versa.)

Subcase 2. n_T is the target of some arc in G_T the source of which is in $G_T \upharpoonright n_{T_i}$ for some $i = 1, \dots, k$, say i_0 . This means that $G_T \upharpoonright n_{T_{i_0}}$ is equal to G_T ; since $G_T \upharpoonright n_{T_{i_0}} \approx G_{T_{i_0}}$, we have $G_T \approx G_{T_{i_0}}$. Hence $r(G_{T_{i_0}}) = r(G_T)$. Since $G_{T_i} \approx G_T \upharpoonright n_i$ which is a subgraph of G_T , $r(G_{T_i}) \leq r(G_T)$ for all i hence $r(G_T) = r(G_{T_{i_0}}) = \mathbf{Max}\{r(G_{T_1}), \dots, r(G_{T_k})\}$ as was to be shown.

Case 3. $e = e_{0,\sigma}(e_1, \dots, e_k)$.

Let $T_i = \mathbf{Val}(e_i)$ for $i = 0, \dots, k$ and $T = \mathbf{Val}(e) = T_0[T_1/\sigma(1), \dots, T_k/\sigma(k)]$. Let (G_i, n_i) be the minimal pointed graph of T_i for $i = 0, \dots, k$. We can assume that the sets of nodes of G_0, \dots, G_k are disjoint. Let G be the pointed ranked graph $(G_0, n_0)[(G_1, n_1)/\sigma(1), \dots, (G_k, n_k)/\sigma(k)]$. Note that for each $i = 1, \dots, k$ there exists in G_0 a node labeled by $\sigma(i)$. It follows that every G_1, \dots, G_k is a subgraph of G and that $\mathbf{SCC}(G) = \mathbf{SCC}(G_0) \cup \dots \cup \mathbf{SCC}(G_k)$ so that $r(G) = \mathbf{Max}\{r(G_0), \dots, r(G_k)\}$. Hence, by induction, we have $r(G_i) = h(e_i)$ for $i = 0, \dots, k$ so that $r(G) = h(e)$.

Let us now show that $r(G_T) \leq r(G)$. It will follow that $h(e) \geq r(G_T)$ hence $h(e) = r(G_T)$ by the hypothesis that e is of minimal star-height and Theorem 2.1.

But

$$\begin{aligned} r(G, n_0) &= T(G_0, n_0)[r(G_1, n_1)/\sigma(1), \dots, r(G_k, n_k)/\sigma(k)] \\ &= T_0[r(T_1/\sigma(1)), \dots, r(T_k/\sigma(k))] \\ &= T. \end{aligned}$$

Since $G_T \upharpoonright n_0 = G$, Proposition 2.5 says that $r(G_T) \leq r(G)$, as was to be shown.

Case 4. $e = *_v(e_0)$ with $e_0 = f(e_1, \dots, e_k)$.

Let $T_0 = \mathbf{Val}(e_0)$ and $T = \mathbf{Val}(e)$. Let (G, n_0) be the pointed ranked graph $*_v(G_{T_0}, n_0)$ (see Definition 1.19). By Proposition 1.20, $T(G, n_0) = T$. It is easy to see that $G \upharpoonright n_0 = G$. It follows from Proposition 2.5 that $r(G_T) \leq r(G)$.

Let n_1, \dots, n_k be the successors of n_0 in G_{T_0} . Let $G_i = G_{T_0} \upharpoonright n_i$. It is clear that $(G_i, n_i) \approx (G_{T_i}, n_i)$, where $T_i = \mathbf{Val}(e_i)$, hence by induction that $h(e_i) = r(G_i)$, hence that

$$h(e) = 1 + \mathbf{Max}\{h(e_1), \dots, h(e_k)\} = 1 + \mathbf{Max}\{r(G_1), \dots, r(G_k)\}.$$

Hence it suffices to show that

$$r(G) \leq 1 + \mathbf{Max}\{r(G_1), \dots, r(G_k)\} \quad (1)$$

in order to conclude that $r(G_T) \leq h(e)$. Let us consider the subgraph $G' = G \setminus n_0$ of G . It is clear that

$$\mathbf{Core}(G \setminus n_0) = \mathbf{Core}(G_{T_0} \setminus n_0),$$

$$\mathbf{SCC}(\mathbf{Core}(G_{T_0} \setminus n_0)) = \mathbf{SCC}(\mathbf{Core}(G_1)) \cup \dots \cup \mathbf{SCC}(\mathbf{Core}(G_k)).$$

Hence,

$$\begin{aligned} r(G \setminus n_0) &= r(\mathbf{Core}(G_{T_0} \setminus n_0)) \\ &= \mathbf{Max}\{r(H) \mid H \in \mathbf{SCC}(\mathbf{Core}(G_i)), i = 1, \dots, k\} \\ &= \mathbf{Max}\{r(H) \mid H \in \mathbf{SCC}(G_i), i = 1, \dots, k\} \\ &= \mathbf{Max}\{r(G_1), \dots, r(G_k)\}. \end{aligned} \quad (2)$$

Since $r(G) \leq 1 + r(G \setminus n_0)$, we get (1) with the help of (2), which concludes the proof. \square

This shows that the (effective) construction of Theorem 3.1 produces an r.e. of minim. star-height defining a given regular tree T .

4. Application to other types of rational expressions

Other types of rational expressions defining regular trees have been introduced by Cousineau [9] and by Bloom and Elgot [2, 12] (see [8] for the relations between the two approaches). They use a single iteration operator defined with respect to a fixed enumerated set of variables $V = \{v_1, v_2, v_3, \dots\}$.

The following class of r.e.'s corresponds to those of Cousineau augmented with a composition operator, defined with respect to the fixed sequence v_1, v_2, \dots instead of arbitrary sequences σ as in $E(F, V)$.

We shall denote $\{v_1, \dots, v_k\}$ by V_k .

4.1. Definition. Let $H(F, V)$ be the set of expressions inductively defined as follows: $e \in H(F, V)$ iff one of the following conditions holds:

- (i) $e \in F \cup V$,
- (ii) $e = f(e_1, \dots, e_k), f \in F_k, k \geq 1, e_1, \dots, e_k \in H(F, V)$,
- (iii) $e = *(e'), e' \in H(F, V)$,
- (iv) $e = e_{0, \sigma}(e_1, \dots, e_k), e_0, \dots, e_k \in H(F, V)$.

The value $\mathbf{Val}(e)$ of e in $H(F, V)$ is defined inductively as follows, according to the four cases:

- (i) $\mathbf{Val}(e) = e$,
- (ii) $\mathbf{Val}(e) = f(\mathbf{Val}(e_1), \dots, \mathbf{Val}(e_k))$,
- (iii) $\mathbf{Val}(e)$ is the unique tree t such that $t = \mathbf{Val}(e')[t/v_1, v_1/v_2, \dots, v_{k-1}/v_k]$ where $\mathbf{Val}(e') \neq v_1$ and $\mathbf{Val}(e') \in M^\infty(F, V_k)$; $\mathbf{Val}(e)$ is undefined if $\mathbf{Val}(e') = v_1$.
- (iv) $\mathbf{Val}(e) = \mathbf{Val}(e_0)[\mathbf{Val}(e_1)/v_1, \dots, \mathbf{Val}(e_k)/v_k]$.

In cases (ii) to (iv), $\mathbf{Val}(e)$ is undefined if any of the required $\mathbf{Val}(e_0), \dots, \mathbf{Val}(e_k), \mathbf{Val}(e')$ is undefined.

The star-height of e in $H(F, V)$ is defined in an obvious way.

4.2. Proposition. For every e in $H(F, V)$ (resp. in $E(F, V)$), a rational expression \bar{e} in $E(F, V)$ (resp. in $H(F, V)$) can be constructed such that $h(\bar{e}) = h(e)$ and $\mathbf{Val}(\bar{e}) = \mathbf{Val}(e)$.

Proof. The translation from $H(F, V)$ into $E(F, V)$ is easy: We do it by following the four cases of Definition 4.1:

- (i) $\bar{e} = e$,
- (ii) $\bar{e} = f(\bar{e}_1, \dots, \bar{e}_k)$,
- (iii) $\bar{e} = (*_{v_1}(\bar{e}'))_{\sigma} (v_1, v_2, \dots, v_{k-1})$, with $\sigma = (v_2, v_3, \dots, v_k)$,
- (iv) $\bar{e} = \bar{e}_{0, \sigma}(\bar{e}_1, \dots, \bar{e}_k)$ with $\sigma = (v_1, \dots, v_k)$.

It is clear that h and \mathbf{Val} are preserved in the transformation. Conversely, let $e \in E(F, V)$:

- (i) If $e \in F \cup V$, then $\bar{e} = e$,
- (ii) if $e = f(e_1, \dots, e_k)$, then $\bar{e} = f(\bar{e}_1, \dots, \bar{e}_k)$,
- (iii) if $e = *_{v_i}(e')$ and $\mathbf{Var}(\mathbf{Val}(e')) \subseteq V_k$, then

$$\bar{e} = *(\bar{e}')_{(v_2, \dots, v_k)} \quad \text{if } i = 1$$

$$\bar{e} = *(\bar{e}')_{(v_{i+1}, v_3, \dots, v_i, v_1, v_{i+2}, \dots, v_{k+1})} \cdot (v_1, \dots, v_{i-1}, v_1, v_{i+1}, \dots, v_k)$$
 if $i \neq 1$.

(iv) if $e = e_{0, \sigma}(e_1, \dots, e_k)$ with $\sigma = (v_{i_1}, \dots, v_{i_k})$, then $\bar{e} = \bar{e}_{0, \sigma}(\bar{e}'_1, \dots, \bar{e}'_k)$, where $l = \mathbf{Max}\{i_1, \dots, i_k\}$ and $e'_i = e_j$ if $i = i_j$ and $e'_i = v_i$ if $i \notin \{i_1, \dots, i_k\}$.

It is clear that $h(\bar{e}) = h(e)$. Let us verify that rule (iii) preserves \mathbf{Val} , i.e., that $\mathbf{Val}(\bar{e}) = \mathbf{Val}(e)$ if we assume inductively that $\mathbf{Val}(\bar{e}') = \mathbf{Val}(e')$.

Case ($i \neq 1$). We have to show that $\mathbf{Val}(\bar{e}) = \mathbf{Val}(e)$, i.e. that $\mathbf{Val}(\bar{e}) = \mathbf{Val}(e')[\mathbf{Val}(\bar{e})/v_i]$. But $\mathbf{Val}(\bar{e}) = t[v_1/v_i]$, where

$$t = (\mathbf{Val}(\bar{e}') [v_{i+1}, v_3, \dots, v_i, v_1, v_{i+2}, \dots, v_{k+1}]) [t'/v_1, v_1/v_2, \dots, v_k/v_{k+1}]$$

$$= \mathbf{Val}(\bar{e}') [v_i, v_2, \dots, v_{i-1}, t, v_{i+1}, \dots, v_k].$$

Hence

$$\begin{aligned} \mathbf{Val}(\bar{e}) &= \mathbf{Val}(\bar{e}')[v_1, v_2, \dots, v_{i-1}, t, v_{i+1}, \dots, v_k][v_1/v_i] \\ &= \mathbf{Val}(\bar{e}')[v_1, \dots, v_{i-1}, \mathbf{Val}(\bar{e}), v_{i+1}, \dots, v_k] \\ &= \mathbf{Val}(\bar{e}')[\mathbf{Val}(\bar{e})/v_i]. \end{aligned}$$

Since we assume that $\mathbf{Val}(\bar{e}') = \mathbf{Val}(e')$, we are done.

The case $i = 1$ is similar and more simple. \square

4.3. Corollary. *Every regular tree T can be defined by a rational expression in $H(F, V)$. The minimal star-height of such a rational expression is the same as for rational expressions in $E(F, V)$, i.e. is $r(G_T)$.*

Regular trees can also be defined by means of iterative theory expressions as recalled in [8, Section 4.7.3]. The star-height problem (where \dagger is considered as the 'star') is trivial since by a result of Elgot [12], every iterative theory expression can be transformed into one using \dagger at most once. This is due to the power of vector-iteration by which several equations can be solved simultaneously.

If one only allows scalar-iterations, one obtains (up to some details) the expressions of $H(F, V)$ and Corollary (4.3) solves the star-height problem.

Application to string rational expressions

We aim to explore the relations between $E(F, V)$ and the classical r.e.'s written with $\cup, \dots, *, \emptyset, \varepsilon$, and the letters of the given alphabet, which define regular languages.

Let us assume that F has no constants and let \bar{F} be the new alphabet $\{f_i \mid f \in F, 1 \leq i \leq \rho(f)\}$.

Every tree in $M^\infty(F, V)$ has a branch-language $\mathbf{Brch}(t) \subseteq \bar{F}^*V$. We only recall from [7, 8] that $\mathbf{Brch}(v) = \{v\}$ and that $\mathbf{Brch}(f(t_1, \dots, t_k)) = f_1 \mathbf{Brch}(t_1) \cup \dots \cup f_k \mathbf{Brch}(t_k)$.

If $t \in M^\infty(F, V_k)$, then $\mathbf{Brch}(t) = L_1 v_1 \cup \dots \cup L_k v_k$ and each L_i is prefix-free.

If t is regular, then the L_i 's are regular. From an r.e. e in $E(F, V_k)$ we shall define a k -tuple $\mathbf{e} = (e_1, \dots, e_k)$ of (string) r.e.'s such that e_i defines L_i .

4.4. Construction. For e in $E(F, V_k)$, we define $\mathbf{e} = (e_1, \dots, e_k)$ by induction on the structure of e .

(i) if $e = v_i$, then $\mathbf{e} = (\emptyset, \emptyset, \varepsilon, \dots, \emptyset)$ with ε at i th position,

(ii) if $e = f(d^1, \dots, d^n)$, then $e_i = f_1 d_i^1 \cup f_2 d_i^2 \cup \dots \cup f_n d_i^n$ and $\mathbf{d}^j = (d_{j,1}^j, \dots, d_{j,k}^j)$ for all $j = 1, \dots, n$,

(iii) if $e = *_{v_i}(e')$, then $e_i = \emptyset$ and $e_j = (e'_j)^* e'_j$ for all $j \neq i$, where $\mathbf{e}' = (e'_1, \dots, e'_k)$,

(iv) if $e = e' \cdot_\sigma (d_1, \dots, d_n)$, where $\sigma = (v_{i_1}, \dots, v_{i_n})$, then $e_i = e'_1 \cdot d'_{1,i} \cup \dots \cup e'_k \cdot d'_{k,i}$, where $\mathbf{e}' = (e'_1, \dots, e'_k)$, $\mathbf{d}_j = (d'_{j,1}, \dots, d'_{j,k})$ for $1 \leq j \leq n$ and with $d'_{j,i} = \emptyset$, $d'_{j,j} = \varepsilon$ when $j \notin \{i_1, i_2, \dots, i_n\}$ and $i \neq j$.

It is clear that $h(e_i) \leq h(e)$.

One can prove that $\mathbf{Brch}(\mathbf{Val}(e)) = \mathbf{Val}(e_1)v_1 \cup \dots \cup \mathbf{Val}(e_k)v_k$ by induction on the structure of e with help of the various lemmas of [7, 8].

The r.e.'s derived from this construction can be simplified with the following rules:

$$\emptyset \cup e \rightarrow e, e \cup \emptyset \rightarrow e, \emptyset.\varepsilon \rightarrow \emptyset, e.\emptyset \rightarrow \emptyset, \varepsilon.e \rightarrow e, e.\varepsilon \rightarrow e, \emptyset^* \rightarrow \varepsilon.$$

Every r.e. can be transformed in a unique reduced expression $\mathbf{red}(e)$ by these rules (*reduced* means that no rule can be applied). Remark that a reduced expression defines the empty language iff it is exactly \emptyset .

4.5. Example. Let $e = e_0.(e_1, e_2)$, with

$$e_0 = *_{v_2}(f(v_1, g(v_2))), \quad e_1 = f(v_1, v_1), \quad e_2 = *_{v_2}(e_3),$$

where

$$e_3 = h(v_1, v_2, *_{v_1}(g(v_1))).$$

We can take $k = 2$. By Construction 4.4 we get

$$e_1 = (f_1.\varepsilon \cup f_2.\varepsilon, \emptyset),$$

which can be reduced into

$$e'_1 = (f_1 \cup f_2, \emptyset).$$

Similarly, one gets for e_0 the reduced

$$e'_0 = ((f_2 g_1)^* f_1, \emptyset).$$

For e_3 one gets, from Construction 4.4,

$$e_3 = (h_1.\varepsilon \cup h_2.\emptyset \cup h_3.\emptyset, h_1.\emptyset \cup h_2.\varepsilon \cup h_3.g_1^*.\emptyset),$$

which can be reduced into

$$e'_3 = (h_1, h_2).$$

From this one gets

$$e'_2 = (h_2^* h_1, \emptyset), \quad e' = ((f_2 g_1)^* f_1 (f_1 \cup f_2), \emptyset).$$

Note that due to the simplifications, the star-height of e' is less than that of e .

In order to characterize the rational expressions constructed in this way, we define a special class of string r.e.'s.

4.6. Definition. Let X be a finite alphabet. For $k \geq 1$, we define the set $D_k(X^*)$ of *k-deterministic rational expressions* over X as the set of k -tuples of r.e.'s $e = (e_1, \dots, e_k)$ by the following inductive definition: e belongs to $D_k(X^*)$ iff one of the following cases holds:

- (i) $e_i = \emptyset$ for all $i = 1, \dots, k$,
- (ii) $e_i = \varepsilon$ for some i and $e_j = \emptyset$ for $j \neq i$.

(iii) $e_i = \mathbf{red}(a_1 e_{1,i} \cup a_2 e_{2,i} \cup \dots \cup a_n e_{n,i})$ for pairwise distinct letters a_1, a_2, \dots, a_n in X and some $e^{(1)}, e^{(2)}, \dots, e^{(n)}$ in $D_k(X^*)$ such that $e^{(j)} = (e_{j,1}, \dots, e_{j,k})$ for $j = 1, \dots, n$.

(iv) $e_i = \mathbf{red}((e'_j)^*(e'_i))$ for $i \neq j$ and $e_j = \emptyset$ for some $j = 1, \dots, k$ and some e' in $D_k(X^*)$.

(v) $e_i = \mathbf{red}(e_{0,1} \cdot e_{1,i} \cup \dots \cup e_{0,k} \cdot e_{k,i})$ for some $e^{(0)}, \dots, e^{(k)}$ in $D_k(X^*)$ such that $e^{(i)} = (e_{i,1}, \dots, e_{i,k})$.

Finally, we define a *deterministic rational expression* (d.r.e.) over X as a component of some e in $D_k(X^*)$ for some k . We denote their set by $D(X^*)$.

4.7. Remark. Whether a given r.e. is a d.r.e. is decidable although not completely trivially. A k -d.r.e. $e = (e_1, \dots, e_k)$ has the star-height $h(e) = \mathbf{Max}\{h(e_1), \dots, h(e_k)\}$.

4.8. Proposition. If e is a rational expression in $E(F, V_k)$, and e is as given in Construction 4.4, then $e' = \mathbf{red}(e)$ is a k -deterministic rational expression over \bar{F} and $h(e') \leq h(e)$.

Proof. The proof clearly follows from the definitions. \square

4.9. Proposition (1) *Deterministic rational expressions are non-ambiguous.*
 (2) *They define prefix-free languages.*

Saying that an r.e. is non-ambiguous means that every word it defines is defined exactly once (see [11]).

We now sketch the proof of the converse of (2) of Proposition 4.9.

Let us assume that X is totally ordered in a fixed way, say $X = \{a_1, \dots, a_m\}$, and let F_X be a ranked alphabet in bijection with $\{\alpha \subseteq X \mid \alpha \neq \emptyset\}$. The rank of the element of F_X corresponding to α is taken as $\mathbf{Card}(\alpha)$. Finally we denote by φ the mapping: $\bar{F}_X \rightarrow X$ associating with α_i (considering α as an element of F_X) its i th element w.r.t. the ordering on X .

By [7, Corollary 5.14] for every prefix-free language $L \subseteq X^*$, there is a unique locally finite tree T_L in $M^x(F_X, V_1)$ such that $\varphi(\mathbf{Brch}(T_L)) = Lv_1$. Furthermore, T_L is regular if L is. And from a r.e. e in $E(F_X, V_k)$ defining T_L , one gets $e = (e_1, \dots, e_k)$ such that $e_2, \dots, e_k = \emptyset$ and $\mathbf{Val}(e_1)v_1 = \mathbf{Brch}(T_L)$. Let $\varphi(e_1)$ be obtained by replacing in e_1 each x in \bar{F}_X by $\varphi(x)$. It is clear that $\mathbf{Val}(\varphi(e_1)) = \varphi(\mathbf{Val}(e_1))$, i.e., that $\varphi(e_1)$ defines L .

We know that $(e_1, \dots, e_k) \in D_k(\bar{F}_X^*)$. Since, for all f in F_X , $\varphi(f_i) \neq \varphi(f_j)$ if $i \neq j$, the k -tuple $(\varphi(e_1), \dots, \varphi(e_k))$ is still in $D_k(X^*)$. Hence, $\varphi(e_1) \in D(X^*)$. These remarks establish the following proposition.

4.10. Proposition. *Every prefix-free regular language is defined by some deterministic rational expression.*

Let us define the *deterministic star-height* $h_d(L)$ of a regular language L as the minimum star-height of a d.r.e. defining L if L is prefix-free and $L\$$ otherwise (where $\$$ is a new symbol). It is easy to see that if L is prefix-free, L and $L\$$ have the same deterministic star-height.

Let G_L be the transition graph associated with the minimal deterministic automaton of L .

4.11. Theorem. *For all regular language L , $h_d(L) = r(G_L)$.*

Proof. Let us first show that $h_d(L) \leq r(G_L)$. Since $r(G_L) = r(G_{L\$})$ we can assume that L is prefix-free.

The tree $T = T_L$ in $M^\times(F_X, V_1)$ canonically associated with L is regular and its minimal graph G_T is isomorphic (up to the labels) to the graph G_L . It follows that $r(G_T) = r(G_L)$. By Theorem 1.12, there exists an r.e. e in $E(F_X, V)$ defining T with star-height $r(G_T)$, and by the proof of Proposition 4.8 there exists a d.r.e. e' of star-height at most $h(e) = r(G_T)$ which defines L . Hence, $h_d(L) \leq r(G_L)$.

Conversely, let $L = \mathbf{Val}(e)$ for some d.r.e. e of star-height n . We shall construct a transition graph G_e of rank at most n , corresponding to some finite deterministic automaton recognizing L .

Since the surjective homomorphism $G_e \rightarrow G_L$ is pathwise, MacNaughton's theorem (see [5, 17]) shows that $r(G_L) \leq r(G_e)$, hence that $r(G_L) \leq n$.

Finally, $r(G_L) \leq h_d(L)$. With every k -d.r.e. $e = (e_1, \dots, e_k)$ we shall associate a pointed directed graph (G_e, u) satisfying the following conditions:

- (i) each arc is labeled by some letter in X .
- (ii) for every node s and every a in X , there is at most one arc with source s and labelled by a ,
- (iii) some nodes are defined as *i-terminal* for some i , $1 \leq i \leq k$, and are not the source of any arc,

(iv) letting $G_{e,i}$ denote the subgraph of G_e consisting of all nodes and arcs belonging to some path from u to some i -terminal node, then $G_e = \bigcup_{1 \leq i \leq k} G_{e,i}$.

From this last condition it is easy to prove that $r(G_e) = \mathbf{Max}\{r(G_{e,i}) \mid 1 \leq i \leq k\}$. It is clear that such a graph G_e is the transition graph of a finite deterministic automaton with final states of k different types. It defines a k -tuple of regular languages $(L_1(G_e, u), \dots, L_k(G_e, u))$ in an obvious way.

We shall construct (G_e, u) so that $L_i(G_e, u) = \mathbf{Val}(e_i)$ (where $e = (e_1, \dots, e_k)$), and $r(G_{e,i}) \leq h(e_i)$ for all $i = 1, \dots, k$.

In the following inductive construction, we assume that $e = (e_1, \dots, e_k)$.

(α) In Definition 4.6(ii) we let G_e be reduced to the node u , considered as i -terminal. In Definition 4.6(i), G_e is the same and u is not i -terminal for any i .

(β) In Definition 4.6(iii) we construct (G_e, u) as follows by using the $(G_{e^{(j)}})$, $u^{(j)}$'s for $1 \leq j \leq n$: we let G_e consist of a new node u and the (disjoint) union of $G_e^{(1)}, \dots, G_e^{(n)}$, augmented with arcs from u to $u^{(j)}$ labelled by a_j , for all $j = 1, \dots, n$.

(γ) In Definition 4.6(v) we construct similarly (G_e, u) from the $(G_{e^{(j)}}, u^{(j)})$'s, $0 \leq j \leq k$, by identifying with $u^{(i)}$ the i -terminal nodes of $G_{e^{(i)}}$ for $1 \leq i \leq k$ and letting $u = u^{(0)}$.

(δ) In Definition 4.6(iv), i.e., if $e_i = \mathbf{red}((e'_j)^* e'_i)$ for $i \neq j$ and $e_j = \emptyset$ for some $j = 1, \dots, k$, then we define (G_e, u) from $(G_{e'}, u')$ by identifying with u' the j -terminal nodes of $G_{e'}$ and letting $u = u'$.

From the construction, it is clear that $L_i(G_e, u) = \mathbf{Val}(e_i)$. One can show by induction on the construction that $r(G_{e_i}) \leq h(e_i)$:

Cases (α), (β), (γ) are easy to establish.

Case (δ):

$$r(G_{e_j}) = 0,$$

$$r(G_{e_i}) \leq \mathbf{Max}\{r(G_{e'_j}) + 1, r(G_{e'_i})\} \quad \text{for } i \neq j.$$

This inequality follows from the remark that by adding one arc to a graph one cannot increase its rank by more than one.

Hence, $r(G_{e_i}) \leq h(e_i)$ if we assume inductively that $r(G_{e'_i'}) \leq h(e'_i')$ for all $i' = 1, \dots, k$.

This construction is applicable to a d.r.e. e defining a language L , by considering it as the k -d.r.e. $(e, \emptyset, \emptyset, \dots, \emptyset)$ for the purpose of induction. It gives us the required graph G_e . \square

4.12. Remark. It is clear that for a regular language L , $h(L) \leq h_d(L)$. The following example is adapted from Cohen and Brzozowski [6] and shows that the inequality may be strict:

$$L = (a \cup ab \cup abbb \cup ba)^* bc, \quad h(L) = 1.$$

The graph G_L is shown in Fig. 2. Its rank is 2. Hence, $h_d(L) > h(L)$.

The same situation holds for the language defined by the r.e. $(ab \cup abbb \cup ba)^* bc$ which is non-ambiguous since $\{ab, abbb, ba\}$ is a code (it is suffix-free).

5. The restricted star-height of regular trees

In this section we establish Theorem 1.14 which says that the restricted star-height of a regular tree is the length of its minimal pointed graph.

We fix F and V exactly as in Sections 1–3.

5.1. Definitions. A *quasi-tree* is a pointed ranked graph (G, n) such that for every node n' in G there exists one and only one cycle-free path from n to n' . This is equivalent to saying that (G, n) is a pointed ranked graph such that

(i) $G = G \upharpoonright n$,

(ii) the set of arcs of G is the disjoint union of two sets A' and A'' such that (N, A') is a tree with root n (N is the set of nodes of G) and every arc a in A''

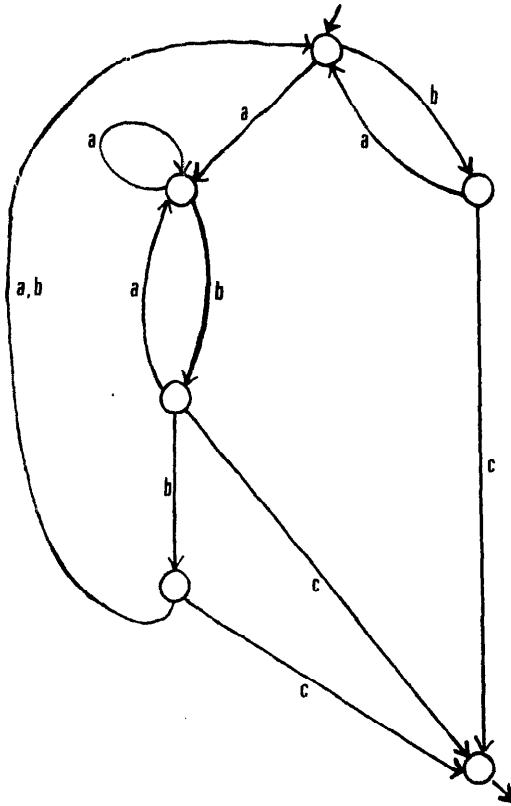


Fig. 2.

has a target which either is equal to the source or is an ancestor of the source with respect to the tree (N, A') .

It is easy to see that the partition of A into $A' \cup A''$ is possible in at most one way. An arc in A'' will be called a *backedge*.

The set C of nodes which are the target of a backedge is a base of G . We shall call it the *canonical base*.

It is easy to see that if (G, n) is reduced to a single node n labeled in $F_0 \cup V$, then it is a quasi-tree, and that the operations on pointed graphs of Definitions 1.17 and 1.19 preserve quasi-trees. It follows that every r.e. in $E_0(F, V)$ denotes a quasi-tree. Let us denote by $Q(e)$ the quasi-tree associated with e . From Propositions 1.18 and 1.20 one immediately gets the following lemma.

5.2. Lemma. *For every e in $E_0(F, V)$, $T(Q(e)) = \mathbf{Val}(e)$.*

Remark that if e' is obtained from e by a renaming of bound variables, then $Q(e') = Q(e)$.

An r.r.e e is in *normal form* if for all subexpressions of e of the form $*_v(e_0)$ then $e_0 = f(e_1, \dots, e_k)$ for some f in F_k and some e_1, \dots, e_k in $E_0(F, V)$ such that v has a free occurrence in at least one of e_1, \dots, e_k .

Equivalently, an r.r.e. is in normal form (is in $NE_0(F, V)$) if and only if either

- (i) $e \in F_0 \cup V$,

- (ii) $e = f(e_1, \dots, e_k)$, with e_1, \dots, e_k in $NE_0(F, V)$,
 (iii) $e = *_v(f(e_1, \dots, e_k))$, with e_1, \dots, e_k in $NE_0(F, V)$ and v has at least one free occurrence in e_1 or \dots or e_k .

5.3. Proposition. (1) Every e in $E_0(F, V)$ can be transformed into an equivalent one e' in normal form, with same quasi-tree and a star-height at most that of e .

(2) Two r.e.'s in normal form defining the same quasi-tree are identical up to a renaming of the bound variables.

(3) Every quasi-tree is defined by some r.r.e.

The proof will be given in Appendix A.

5.4. Remark. If $(G, n) = Q(e)$ for e in $E_0(F, V)$, then $h(e) = l_C(G, n)$, where C is the canonical base of G .

We shall establish Theorem 1.14 saying that for every regular tree T , $h_0(T) = l(G_T, n_T)$ as we did for Theorem 1.12, that is, by first constructing an r.e. from a pointed graph (G, n) and then showing that it is minimal when applied to the minimal pointed graph of some tree.

The following proposition gives the construction and yields the inequality $h_0(T) \leq l(G_T, n_T)$.

5.5. Proposition. Let (G, n) be a pointed ranked graph with base B . One can construct a quasi-tree (H, m) such that $T(H, m) = T(G, n)$ and $l(H, m) \leq l_B(G, n)$.

Proof. Together with (H, m) one defines a homomorphism $\varphi: H \rightarrow G$ such that $\varphi(m) = n$ (this insures $T(H, m) = T(G, n)$) and the canonical base C of (H, m) ; we shall have $l_C(H, m) \leq l_B(G, n)$ (insuring $l(H, m) \leq l_B(G, n)$).

Since we can always replace G by $G \uparrow n$ and B by $B' = B \cap (G \uparrow n)$ so that $T(G \uparrow n, n) = T(G, n)$ and $l_{B'}(G \uparrow n, n) \leq l_B(G, n)$ (it is clear that B' is a base of $G \uparrow n$), we can assume that $G \uparrow n = G$.

We do the proof by induction on the integer $m(G, n, B) = \sum \{|\gamma| \mid \gamma \in \Gamma\}$, where $|\gamma|$ is the number of nodes of a cycle-free path γ and Γ is the (finite) set of cycle-free paths from n to the elements of B . (If $n \in B$, we consider (n) as a path in Γ , and count it for 1.)

Letting $G = (N, \mathbf{Suc}, \mathbf{Lab})$ (such that $G \uparrow n = G$) we have 4 cases to consider.

Case 1. $\mathbf{Lab}(n) \in F_0 \cup V$.

Then we take $(H, m) = (G, n)$, $C = \emptyset$ φ the identity.

In all other cases we assume $\mathbf{Lab}(n) = f \in F_k$, $k \geq 1$ and $\mathbf{Suc}(n) = (n_1, \dots, n_k)$.

Case 2. n is not the target of any arc.

Let $G_i = G \uparrow n_i$, $B_i = B \cap (G \uparrow n_i)$. It is clear that $m(G_i, n_i, B_i) < m(G, n, B)$ hence we can assume inductively that we already have (H_i, m_i) , $\varphi_i: H_i \rightarrow G_i$, sending m_i onto n_i , and C_i such that $l_{C_i}(H_i, m_i) \leq l_{B_i}(G_i, n_i)$. We can assume that H_1, \dots, H_k

have disjoint sets of nodes. Then we take $(H, m) = f((H_1, m_1), \dots, (H_k, m_k))$ for some new node m , $\varphi(m) = n$, $\varphi(m') = \varphi_i(m')$ for a node m' in H_i , $C = C_1 \cup \dots \cup C_k$.

It is clear that C is the canonical base of H and that

$$l_C(H, m) = \mathbf{Max}\{l_{C_i}(H_i, m_i) \mid 1 \leq i \leq k\}.$$

Similarly,

$$l_B(G, n) = \mathbf{Max}\{l_{B_i}(G_i, n_i) \mid 1 \leq i \leq k\}.$$

Since by induction we assume that $l_{C_i}(H_i, m_i) \leq l_{B_i}(G_i, n_i)$ we get $l_C(H, m) \leq l_B(G, n)$ as wanted.

Case 3. $n \in B$ and n is the target of some arcs.

Let v be a new variable (not appearing in G).

Let G_1 be obtained from G by adding a new node s labelled by v and letting s be the target of every arc having target n in G , so that $(G, n) = *_v(G_1, n)$.

Let $B_1 = B - \{n\}$.

It is easy to check from the definitions that B_1 is a base of G_1 , that $l_{B_1}(G_1, n) = l_B(G, n) - 1$.

The set of paths Γ_1 is exactly Γ minus the one-node path (n) hence $m(G_1, n, B_1) = m(G, n, B) - 1$. One can assume by induction that we have (H_1, m) , φ_1 , C_1 associated with (G_1, n) .

One then let $(H, m) = *_v(H_1, m)$, φ be the restriction of φ_1 to the nodes of H (i.e., the nodes of H_1 not labelled by v), $C = C_1 \cup \{m\}$. It is clear that

$$l_C(H, m) = l_{C_1}(H_1, m) + 1 \leq l_{B_1}(G_1, m) + 1 = l_B(G, n),$$

that φ is a homomorphism and C is the canonical base of H .

Case 4. $n \notin B$ and n is the target of some arcs.

Let s be a new node. Let G_1 be obtained from G by:

- (i) the addition of node s , labelled by f ,
- (ii) the addition of the arcs from s to nodes in G , in such a way that $\mathbf{Suc}(s) = (n_1, \dots, n_k)$ in G_1 ,
- (iii) defining s as the new target of all arcs having target n in G .

The homomorphism $\pi: G_1 \rightarrow G$ sending s onto n and all other nodes onto themselves shows that $T(G_1, n) = T(G, n)$.

Let us define $B_1 = B$. This set is a base of G_1 and $l_{B_1}(G_1, n) = l_B(G, n)$.

If $\gamma_1 = (n, m_1, m_2, \dots, m_l)$ is a B -simple path of G , then $\varphi(\gamma_1) = (n, \varphi(m_1), \varphi(m_2), \dots, \varphi(m_l))$ is B -simple since φ is injective on B . Hence, $\mathbf{Card}(B_1 \cap \gamma_1) = \mathbf{Card}(B \cap \varphi(\gamma_1))$. It follows that $l_{B_1}(G_1, n) \leq l_B(G, n)$.

Conversely, any path $\gamma = (n, m_1, \dots, m_l)$ in G is $\varphi(\gamma_1)$ for a unique path $\gamma_1 = (n, m_1, \dots, m'_l)$ in G_1 and $\mathbf{Card}(B_1 \cap \gamma_1) = \mathbf{Card}(B \cap \gamma)$, hence $l_B(G, n) \leq l_{B_1}(G_1, n)$.

Then $\Gamma_1 = \Gamma$ and $m(G_1, n, B_1) = m(G, n, B)$ but Case 2 is applicable to (G_1, n, B_1) since n is not the target of any arc in G_1 . Hence, by the argument of the Case 2 and induction, one can assume that one has (H, m) , φ_1 , C associated with (G_1, n)

and B_1 . Setting $\varphi = \pi \circ \varphi_1$, the triple (H, m) , φ , C is the required one. In particular, $T(H, m) = T(G, n)$ (due to the existence of φ), $l_C(H, m) \leq l_{B_1}(G_1, n) = l_B(G, n)$. \square

5.6. Corollary. *For every regular tree T , one can construct an r.e. in $E_0(F, V)$ defining T and such that $h(e) \leq l(G_T, n_T)$. Hence, $h_0(T) \leq l(G_T, n_T)$.*

Proof. Let B be a base of G_T for which $l_B(G_T, n_T) = l(G_T, n_T)$. The quasi-tree (H, m) associated with G_T, n_T, B by Proposition 5.5 corresponds to an r.e. e in $NE_0(F, V)$ of star-height $l_C(H, m) \leq l_B(G_T, n_T)$. \square

5.7. Remarks. As for the construction of Proposition 2.6, the present one is effective provided one knows a base B for which $l_B(G_T, n_T)$ is minimal. It can be found by brute force enumeration but an efficient algorithm remains to be found. This remark applies also to the computation of $l(G_T, n_T)$.

5.8. Proposition. *Let $e \in E_0(F, V)$ be an r.e. of minimal star-height defining T . One can define a base B of G_T such that $h(e) \geq l_B(G_T, n_T)$. Hence, $h_0(T) \geq l(G_T, n_T)$.*

Proof. Since by Proposition 5.3 one can transform an r.e. e in $E_0(F, V)$ into an equivalent one in $NE_0(F, V)$ without increasing the star-height, one can assume that e is in $NE_0(F, V)$.

Let (H, m) be the associated quasi-tree with canonical base C . Let $\varphi: (H, m) \rightarrow (G_T, n_T)$ be the unique homomorphism (see Proposition 1.9). It is surjective. Let $B = \varphi(C)$. We have to verify that B is a base.

Recall from the proof of Proposition 2.5 that for every path γ from n' to n'' in G , for every s' in H such that $\varphi(s') = n'$, there is a unique path in H starting at s' having the image γ by φ ; it will be denoted by $\varphi_s^{-1}(\gamma)$.

Let now γ be a cycle, say, from n_1 to n_1 in G_T . Let s be an arbitrary node of H such that $\varphi(s) = n_1$ and consider the paths $\varphi_s^{-1}(\gamma^k)$ for all $k \geq 1$. There exist k and $k + k'$ such that $\varphi_s^{-1}(\gamma^k)$ and $\varphi_s^{-1}(\gamma^{k+k'})$ have the same target s' . Hence, there exists a cycle from s' to s' in H with image $\gamma^{k'}$ by φ .

This cycle has some node s'' in C . Its image $\varphi(s'')$ belongs to $\gamma^{k'}$ hence to γ and also to B . Hence B is a base of G_T .

Let us now show that $l_C(H, m) \geq l_B(G_T, n_T)$.

Let $\gamma: (n_l = n_0, n_1, \dots, n_k)$ be a B -simple path in G_T and $B' = B \cap \gamma$. Let $\varphi_m^{-1}(\gamma) = (m = m_0, m_1, \dots, m_k)$. If $i \neq j$ and $m_i = m_j \in C$, then $n_i = n_j \in B$ contradicting the hypothesis. Hence, $\varphi_m^{-1}(\gamma)$ is C -simple.

It is clear that $B' = \varphi(C \cap \varphi_m^{-1}(\gamma))$, hence,

$$\text{Card}(B') \leq \text{Card}(C \cap \varphi_m^{-1}(\gamma)) \leq l_C(H, m).$$

Hence, $l_B(G_T, n_T) \leq l_C(H, m)$. \square

Proposition 5.8 and Corollary 5.6 give us Theorem 1.14, i.e., the equality $h_0(T) = l(G_T, n_T)$.

We now show that this result applies to the rational expressions defined by Cousineau [9] and to the class of restricted scalar (iterative theory) expressions defined by Courcelle [8].

With the notations of the present paper, these expressions correspond to the subset $H_0(F, V)$ of $H(F, V)$ consisting of r.e.'s of $H(F, V)$ which do not use the composition operator $.(, \dots,)$.

All that we need is the following analogue of Proposition 4.2:

5.9. Proposition. *For every r.r.e. in $H_0(F, V)$ (resp. in $E_0(F, V)$) one can construct an r.r.e. \bar{e} in $E_0(F, V)$ (resp. in $H_0(F, V)$) which defines the same tree and has no larger star-height.*

Proof. Let $e \in H_0(F, V)$. We define \bar{e} inductively as follows by assuming that no variable is both bound and free in \bar{e} (bound variables are renamed if necessary):

$$\begin{aligned} \bar{e} &= e && \text{if } e \in F_0 \cup V, \\ \bar{e} &= f(\bar{e}_1, \dots, \bar{e}_k) && \text{if } e = f(e_1, \dots, e_k), \\ \bar{e} &= *_{v_l}(\bar{e}'[v_l/v_1, v_1/v_2, v_2/v_3, \dots, v_{k-1}/v_k]) && \text{if } e = *(e'), \end{aligned}$$

where k is such that $\mathbf{Fvar}(e') \subseteq \{v_1, \dots, v_k\}$ and l is large enough so that v_l does not appear in \bar{e}' .

Let us only verify that $\mathbf{Val}(\bar{e}) = \mathbf{Val}(e)$ if $\mathbf{Val}(\bar{e}') = \mathbf{Val}(e')$. It suffices to show that

$$\mathbf{Val}(\bar{e}) = \mathbf{Val}(e')[\mathbf{Val}(\bar{e})/v_1, v_1/v_2, \dots, v_{k-1}/v_k]. \tag{3}$$

But from the definition

$$\begin{aligned} \mathbf{Val}(\bar{e}) &= \mathbf{Val}(\bar{e}'[v_l/v_1, v_1/v_2, \dots, v_{k-1}/v_k])[\mathbf{Val}(\bar{e})/v_l] \\ &= \mathbf{Val}(\bar{e}')[v_l/v_1, v_1/v_2, \dots, v_{k-1}/v_k][\mathbf{Val}(\bar{e})/v_l] \\ &= \mathbf{Val}(\bar{e}')[\mathbf{Val}(\bar{e})/v_1, v_1/v_2, \dots, v_{k-1}/v_k], \end{aligned}$$

which gives (3) since we assume that $\mathbf{Val}(\bar{e}') = \mathbf{Val}(e')$.

The other verifications are trivial.

Let us now start from e in $E_0(F, V)$. We can assume that no variable in e is both bound and free, nor is bound twice. Let k be large enough so that $e \in E_0(F, V_k)$.

By induction on the structure of e , we define \bar{e} in $H_0(F, V)$ for all e in $E_0(F, V_k)$ satisfying the above condition on the bound variables.

$$\begin{aligned} \bar{e} &= e && \text{if } e \in F_0 \cup V, \\ \bar{e} &= f(\bar{e}_1, \dots, \bar{e}_k) && \text{if } e = f(e_1, \dots, e_k), \\ \bar{e} &= *(e'') && \text{if } e = *_{v_l}(e'), \end{aligned}$$

and where $e'' = \mathbf{comp}(\bar{e}', v_2, v_3, \dots, v_k, v_1, v_{l+2}, \dots, v_{k+1})$.

Here we use an operation on $H_0(F, V)$ introduced in [8, Definition 4.5.4] which associates with an $(n+1)$ -tuple of r.r.e.'s (e, e_1, \dots, e_n) an r.r.e. $e' = \mathbf{comp}(e, e_1, \dots, e_n)$ such that

$$\mathbf{Val}(e') = \mathbf{Val}(e)[\mathbf{Val}(e_1)/v_1, \dots, \mathbf{Val}(e_n)/v_n].$$

This operation allows to eliminate the composition operator $.(, \dots, .)$ from r.e.'s, i.e., to translate in $H_0(F, V)$ a r.e. of $H(F, V)$.

Let us verify that $\mathbf{Val}(\bar{e}) = \mathbf{Val}(e)$ by assuming that $\mathbf{Val}(\bar{e}') = \mathbf{Val}(e')$ and the fundamental property of **comp** recalled above.

We have to verify that

$$\mathbf{Val}(e) = \mathbf{Val}(e'')[\mathbf{Val}(e)/v_1, v_1/v_2, \dots, v_k/v_{k+1}].$$

But this amounts to verify

$$\begin{aligned} \mathbf{Val}(e) = \mathbf{Val}(\bar{e}') & [v_2/v_1, \dots, v_l/v_{l-1}, v_1/v_l, v_{l+2}/v_{l+1}, \dots, v_{k+1}/v_k] \\ & [\mathbf{Val}(e)/v_1, v_1/v_2, \dots, v_k/v_{k+1}]; \end{aligned}$$

equivalently

$$\mathbf{Val}(e) = \mathbf{Val}(\bar{e}') [v_1/v_1, \dots, v_{l-1}/v_{l-1}, \mathbf{Val}(e)/v_l, v_{l+1}/v_{l+1}, \dots, v_k/v_k],$$

and this holds from the hypothesis that $\mathbf{Val}(\bar{e}') = \mathbf{Val}(e')$ and $e = *_{v_l}(e')$.

There only remains to establish that $h(e'') \leq h(e')$. We show that $h(e'') \leq h(\bar{e}')$ in the above construction by looking at the definition of **comp** that we recall from [8]; $e' = \mathbf{comp}(e, e_1, \dots, e_n)$ is the following r.e.:

- (i) $e' = e_i$ if $e = v_i$, $1 \leq i \leq n$,
- (ii) $e' = e$ if $e = v_i$, $i > n$ or if $e \in F_0$,
- (iii) $e' = f(e'_1, \dots, e'_k)$ if $e = f(d_1, \dots, d_k)$ and $e'_i = \mathbf{comp}(d_i, e_1, \dots, e_n)$, $i = 1, \dots, k$
- (iv) $e' = *(\mathbf{comp}(e'', v_1, d_1, \dots, d_n))$ if $e = *(e'')$ and where $d_i = \mathbf{comp}(e_i, v_2, v_3, \dots, v_{i+1})$ for $i = 1, \dots, n$ and l is large enough so that $\mathbf{Fvar}(e_i) \subseteq \{v_1, \dots, v_l\}$.

The result will follow from the following claim.

Claim. If $e_1, \dots, e_n \in V$, then $h(\mathbf{comp}(e, e_1, \dots, e_n)) \leq h(e)$.

We prove it by induction on the structure of e .

Cases (i)–(iii) are clear.

In case (iv) it is also clear that the d_i 's belong to V . Hence, by induction, $h(\mathbf{comp}(e'', v_1, d_1, \dots, d_n)) \leq h(e'')$ and $h(e') \leq h(e)$. \square

This concludes the proof of Proposition 5.9 and shows that $h_0(T) = l(G_T, n_T)$ is also the minimal star-height of an r.r.e. in $H_0(F, V)$ defining T .

5.10. Remark. A family of rational expressions for languages could be derived from $E_0(F, V)$ as we did in Section 4 from $E(F, V)$ and a result analogous to Theorem 4.11 could be derived from Theorem 1.14.

5.11. Example. We shall use the system Σ of Example 2.8. Due to the presence of loops (see Fig. 1(a)) the only base of G is $\{x, y, z, t\}$.

Note then that

$$l(G, x) = 4, \quad l(G, y) = 3, \quad l(G, z) = 3, \quad l(G, t) = 1,$$

so that $h_0(X) = 4, h_0(Y) = h_0(Z) = 3$ whereas $h(X) = h(Y) = h(Z) = 2$. Let us construct an expression in $E_0(F, V)$ for defining Y . We shall define it as

$$e_0 = *_v(g(v, e_1, e_2)),$$

where e_1 should define the tree $T(G''', x)$ and e_2 the tree $T(G''', z)$, where G''' is shown in Fig. 3 (compare with Fig. 1(c)).

We can take

$$e_1 = *_x(f(x, v)), \quad e_2 = *_z(g(z, v, e_3)),$$

where e_3 should define $T(G''', t)$. Clearly we can take

$$e_3 = *_t(h(t, t)).$$

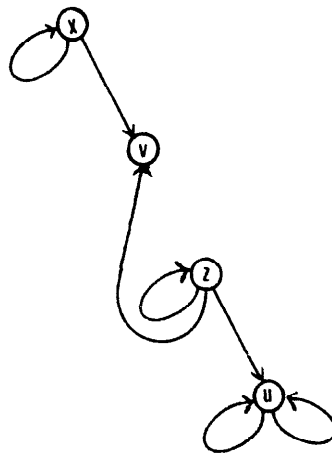


Fig. 3.

5.12. Example. This example shows that the canonical base C of a quasi-tree (G, n) is not necessarily the 'best' one, i.e., that we can have $l(G, n) < l_C(G, n)$. Let (G, x) be the quasi-tree $Q(e)$ associated with $e = *_x(f(*_y(g(y, h(a, x))))))$ with nodes denoted by x, y, z, t in an obvious way, labeled with f, g, h, a , and represented in Fig. 4(a).

Its canonical base is $\{x, y\}$ and $l_C(G, x) = 2$ but $\{y\}$ is also a base giving $l(G, x) = 1$. From it one can construct the r.r.c.

$$e' = f(*_y(g(y, h(a, f(y))))),$$

equivalent to e and of smaller star-height. The quasi-tree of e' is shown in Fig. 4(b).

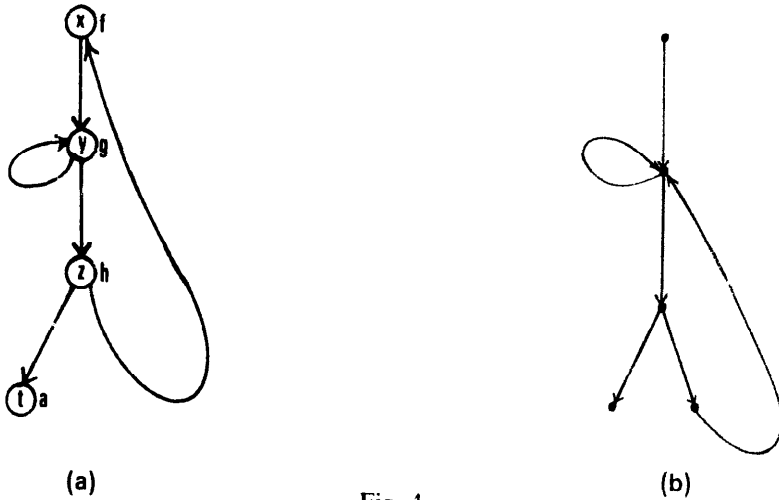


Fig. 4.

5.13. Example. Consider the pointed graph $(G, 1)$ where G is shown in Fig. 5. The sets $B = \{1, 2, 3\}$ and $B' = \{4, 5\}$ are two minimal bases of G . We have $l_{B'}(G, 1) = 2$ and $l_B(G, 1) = 3$ (due to the path 1, 4, 2, 5, 3).

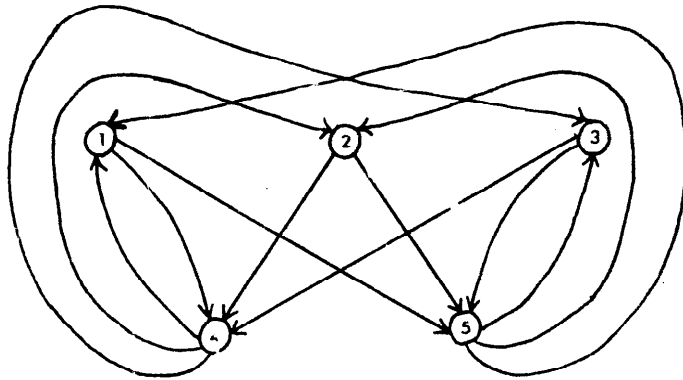


Fig. 5.

Appendix A

This appendix is devoted to the proofs of Propositions 3.2 and 5.3.

A.1. Definition. We first define a set of rewriting rules on $E(F, V)$ that will be used to simplify an r.e. in a certain way.

Let R be the union of the following sets of rules (R1) to (R3), where e, e_1, \dots, e_k are in $E(F, V)$:

$$(R1) \quad v_{\cdot\sigma}(e_1, \dots, e_k) \rightarrow e_i \quad \text{if } v = \sigma(i),$$

$$(R2) \quad e_{\cdot\sigma}(e_1, \dots, e_k) \rightarrow e_{\cdot\sigma'}(e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_k),$$

where $\sigma' = (\sigma(1), \dots, \sigma(i-1), \sigma(i+1), \dots, \sigma(k))$,
if $\sigma(i) \notin \mathbf{Var}(\mathbf{Val}(e))$,

$$(R3) \quad *_v(e) \rightarrow e \quad \text{if } v \notin \mathbf{Var}(\mathbf{Val}(e)).$$

The notation $e \rightarrow_R \bar{e}$ means that one can find a subexpression e' of e which is the left part of some rule in R . Letting \bar{e}' be the corresponding right part, \bar{e} is obtained by the replacement of e' by \bar{e}' at one of its occurrences in e .

This corresponds to one step of simplification of e .

It is clear that $\mathbf{Val}(\bar{e}) = \mathbf{Val}(e)$ and that $h(\bar{e}) \leq h(e)$ for any rule $e \rightarrow \bar{e}$ in R . It follows, in particular from Definition 1.3, that the same holds for e, \bar{e} such that $e \rightarrow_R \bar{e}$.

Since each rule of R reduces the length of the expression to which it applies, \rightarrow_R is Noetherian. It is easy to prove that it is also locally confluent, hence confluent (see [15]). Hence, every r.e. e has a unique normal form w.r.t. \rightarrow_R , that we shall denote by $\mathbf{simpl}(e)$. Our previous remarks have established the following lemma.

A.2. Lemma. *For every r.e. e $\mathbf{simpl}(e)$ is equivalent to e and $h(\mathbf{simpl}(e)) \leq h(e)$.*

We denote by $\mathbf{SE}(F, V)$ the set $\{\mathbf{simpl}(e) \mid e \in E(F, V)\}$. It would be easy to prove by induction on the structure of e that for all e in $\mathbf{SE}(F, V)$, $\mathbf{Fvar}(e) = \mathbf{Var}(\mathbf{Val}(e))$, whereas only \supseteq holds in the general case.

We now define a new type of simplification based on the remarks that $*_v(*_w(e))$ is equivalent to $*_v(e[v/w])$, and that $*_v((*_w(e))_{.\sigma}(e_1, e_2))_{.\tau}(e_3, e_4)$ is equivalent to $*_v((e[v/w])_{.\sigma}(e_1, e_2))_{.\tau}(e_3, e_4)$ (provided v does not appear in σ and τ).

We include all these situations by defining an operator q_v such that $*_v(q_v(e))$ is equivalent to $*_v(e)$ and $h(q_v(e)) \leq h(e)$.

Let us recall that $e[v/\dot{w}]$ denotes the substitution of v for all free occurrences of w in e .

A.3. Definition. For every v in V let q_v be the following operator defined on $\mathbf{SE}(F, V) - V$:

- (i) $q_v(*_w(e)) = q_v(e)[v/w]$,
- (ii) $q_v(e_{.\sigma}(e_1, \dots, e_k)) = q_v(e)_{.\sigma}(e_1, \dots, e_k)$,
- (iii) $q_v(f(e_1, \dots, e_k)) = f(e_1, \dots, e_k)$,

where in (i) we allow $w = v$, in (ii) we assume that $v \notin \sigma$ (some renaming is done in e and σ if necessary) and in (iii) we allow $k = 0$.

A.4. Lemma. (1) *For e in $\mathbf{SE}(F, V) - V$ such that $v \in \mathbf{Fvar}(e)$, $q_v(e)$ is defined, belongs to $\mathbf{SE}(F, V)$ and $h(q_v(e)) \leq h(e)$.*

(2) *Let $*_v(e) \in \mathbf{SE}(F, V)$. Then $*_v(q_v(e))$ is equivalent to $*_v(e)$ and belongs to $\mathbf{SE}(F, V)$.*

Proof. (1) It is clear that the computation of $q_v(e)$ terminates for all e in $\mathbf{SE}(F, V) - V$.

By using an induction on the computation, it is not difficult to prove that $q_v(e) \in \text{SE}(F, V)$ by using the auxiliary assertion that $\mathbf{Fvar}(q_v(e)) = \mathbf{Fvar}(e)$.

Case (i). $\mathbf{Fvar}(e) = \{v, w, x, y, \dots, t\} = \mathbf{Fvar}(q_v(e))$ by induction;

$$\mathbf{Fvar}(q_v(e)[v/w]) = \{v, x, y, \dots, t\} = \mathbf{Fvar}(*_w(e)).$$

It is easy to see that if e' is in $\text{SE}(F, V)$, then so is $e'[v/w]$, hence $q_v(*_w(e)) \in \text{SE}(F, V)$ if $q_v(e) \in E(F, V)$.

Case (ii). From the assumption that $e_{\cdot\sigma}(e_1, \dots, e_k) \in \text{SE}(F, V)$ we have $\mathbf{Fvar}(e) = \{v, \sigma(1), \dots, \sigma(k), x, y, \dots, t\}$ and $\mathbf{Fvar}(q_v(e))$ is the same.

Now

$$\begin{aligned} \mathbf{Fvar}(e_{\cdot\sigma}(e_1, \dots, e_k)) &= \{v, x, y, \dots, t\} \cup \bigcup \{\mathbf{Fvar}(e_i) \mid 1 \leq i \leq k\} \\ &= \mathbf{Fvar}(q_v(e)_{\cdot\sigma}(e_1, \dots, e_k)). \end{aligned}$$

Hence, $\mathbf{Fvar}(e_{\cdot\sigma}(e_1, \dots, e_k)) = \mathbf{Fvar}(q_v(e_{\cdot\sigma}(e_1, \dots, e_k)))$.

Since $e_1, \dots, e_k, q_v(e) \in \text{SE}(F, V)$, the only possibilities for $q_v(e)_{\cdot\sigma}(e_1, \dots, e_k)$ not to be in $\text{SE}(F, V)$ are that rules (R1) or (R2) are applicable at the top level. Since $q_v(e)$ never belongs to V , the first case is excluded. Since

$$\mathbf{Var}(\mathbf{Val}(e)) = \mathbf{Fvar}(e) = \mathbf{Fvar}(q_v(e)) = \mathbf{Var}(\mathbf{Val}(q_v(e))),$$

the second case would imply that a similar rule would be applicable to $e_{\cdot\sigma}(e_1, \dots, e_k)$ contradicting the initial assumption.

Hence, $q_v(e_{\cdot\sigma}(e_1, \dots, e_k)) \in \text{SE}(F, V)$.

Case (iii). Trivial.

(2) Since

$$\mathbf{Var}(\mathbf{Val}(q_v(e))) = \mathbf{Fvar}(q_v(e)) = \mathbf{Fvar}(e) = \mathbf{Var}(\mathbf{Val}(e)),$$

no rule of (R3) is applicable to $*_v(q_v(e))$ if $*_v(e) \in \text{SE}(F, V)$.

To prove that $*_v(q_v(e))$ is equivalent to $*_v(e)$ we also use an induction on the computation of $q_v(e)$.

Case (i). We must show that

$$*_v(*_w(e)) \equiv *_v(q_v(e)[v/w])$$

from the hypothesis that

$$*_v(q_v(e)) \equiv *_v(e),$$

but this follows from Lemma 1.22.

Case (ii). We must show that

$$*_v(e_{\cdot\sigma}(e_1, \dots, e_k)) \equiv *_v(q_v(e)_{\cdot\sigma}(e_1, \dots, e_k))$$

from the hypothesis that

$$*_v(q_v(e)) \equiv *_v(e).$$

But this follows from Lemmas 1.21 and 1.22 by some calculations:

$$\begin{aligned}
 *_v(e_{\sigma}(e_1, \dots, e_k)) &\equiv *_w(*_v(e_{\sigma}(e_1[w/v], \dots, e_k[w/v]))) \\
 &\equiv *_w((*_v(e))_{\sigma}(e_1[w/v], \dots, e_k[w/v])) \\
 &\equiv *_w((*_v(q_v(e))_{\sigma}(e_1[w/v], \dots, e_k[w/v]))) \\
 &\equiv *_v(q_v(e)_{\sigma}(e_1, \dots, e_k)),
 \end{aligned}$$

where w does not occur in e, e_1, \dots, e_k .

Case (iii). Trivial since $q_v(e) = e$. \square

In the normal form we wish to get in Proposition 3.2 we require that in every r.e. of form $*_v(e_0)$, the i e. e_0 is of the form $f(e_1, \dots, e_k)$.

The operator q_v helps by putting together several iterations but is insufficient to transform $*_v((f(e_1, e_2)_{\sigma}(e_3, e_4))_{\tau}(e_5, e_6))$ into

$$*_v(f((e_{1\sigma}(e_3, e_4))_{\tau}(e_5, e_6), (e_{2\sigma}(e_3, e_4))_{\tau}(e_5, e_6))).$$

This will be the job of the operator p defined below.

A.5. Definition. Let $p: SE(F, V) \rightarrow SE(F, V)$ be the following operator:

- (i) $p(v)$ is undefined,
- (ii) $p(*_v(e))$ is undefined,
- (iii) $p(f(e_1, \dots, e_k)) = f(e_1, \dots, e_k)$ with $k \geq 0$,
- (iv) $p(e_{\sigma}(e_1, \dots, e_k)) = f(e'_1, \dots, e'_k)$ if $p(e)$ is defined and equal to $f(e''_1, \dots, e''_k)$ and $e'_i = \mathbf{simple}(e''_{i\sigma}(e_1, \dots, e_k))$ for all i and is undefined if $p(e)$ is.

We shall denote by $SE_p(F, V)$ the set of r.e.'s in $SE(F, V)$, for which p is defined.

A.6. Lemma. Let $e \in SE_p(F, V)$. Then $p(e)$ belongs to $SE(F, V)$, is equivalent to e and $h(p(e)) \leq h(e)$. We have $p(e) = f(e_1, \dots, e_k)$ for some f in F_k , $k \geq 0$, and $|e_i| < |e|$ for all $i = 1, \dots, k$. If $e \notin V$, then $p(q_v(e))$ is defined.

Proof. The first three assertions are easy (by induction). Let us prove that if $p(e) = f(e_1, \dots, e_k)$, then $|e_i| < |e|$.

This is clear for case (iii).

For case (iv) we can assume that $|e''_i| < |e|$ for all i . Hence,

$$|e'_i| \leq |e''_i| + |e_1| + \dots + |e_k| < |e| + |e_1| + \dots + |e_k| = |e_{\sigma}(e_1, \dots, e_k)|.$$

That $p(q_v(e))$ is defined for e in $SE(F, V) - V$ can be shown by induction on the structure of e . \square

A.7. Definition. An r.e. e in $F(E, V)$ is *strongly minimal* if it is minimal (i.e., of minimal star-height) and there is no subexpression e' of e that could be replaced in e by \bar{e}' giving \bar{e} that $h(\bar{e}') < h(e')$ and $\mathbf{Val}(\bar{e}) = \mathbf{Val}(e)$.

This implies in particular that every subexpression of e is minimal. Here is an example of an r.e. which is minimal together with all its subexpressions:

$$e = *_v((*_w(f(v, w, x)))_x *_y(g(y, v, a))).$$

It is not strongly minimal since $*_w(f(v, w, x))$ can be replaced in e by $f(v, v, x)$ giving the equivalent

$$\bar{e} = *_v(f(v, v, x)_x *_y(g(y, v, a))).$$

A.8. Lemma. *Every r.e. is equivalent to some minimal r.e. belonging to $SE(F, V)$.*

Proof. Let e be minimal. Then $\mathbf{simpl}(e)$ is equivalent to e and $h(\mathbf{simpl}(e)) \leq h(e)$ by Lemma A.2, hence we have an equality and $\mathbf{simpl}(e)$ is minimal. \square

A.9. Definition. An r.e. is in *normal form* if it is in $SE(F, V)$ and for all its subexpressions of form $*_i(e_0)$, the r.e. e_0 is of the form $f(e_1, \dots, e_k)$ for some f in F_k , $k \geq i$ and some r.e. e_1, \dots, e_k .

All our efforts converge to the following proposition.

A.10. Proposition. *Every r.e. is equivalent to some strongly minimal r.e. in normal form.*

Proof. By Lemma A.8 we can start from e in $SE(F, V)$ of minimal star-height. We show the existence of the desired $\mathbf{norm}(e)$ by induction on $(h(e), |e|)$ w.r.t. the lexicographical ordering.

So let $e \in SE(F, V)$ be of minimal star-height. We consider several cases.

Case 1. $e \in F_0 \cup V$. Obviously, $\mathbf{norm}(e) = e$.

Case 2. $e = f(e_1, \dots, e_k)$. For every $i = 1, \dots, k$ let e'_i be equal to e_i if e_i is of minimal star-height and be any equivalent r.e. in $SE(F, V)$ of minimal star-height otherwise.

In both cases, $(h(e'_i), |e'_i|) < (h(e), |e|)$ so that by induction we can assume the existence of $\mathbf{norm}(e'_i)$, equivalent to e_i , strongly minimal and in normal form. So let $\mathbf{norm}(e) = f(\mathbf{norm}(e'_1), \dots, \mathbf{norm}(e'_k))$. Then $\mathbf{norm}(e)$ is equivalent to e , $h(\mathbf{norm}(e)) \leq h(e)$, hence the equality holds since e is assumed minimal. It follows that $\mathbf{norm}(e)$ is minimal. It is strongly minimal and in normal form since $\mathbf{norm}(e'_1), \dots, \mathbf{norm}(e'_k)$ are.

Case 3. $e = e_{0,\sigma}(e_1, \dots, e_k)$ with $k \geq 1$. The argument is the same with e'_i for $i = 0, \dots, k$. One gets

$$\mathbf{norm}(e) = \mathbf{norm}(e'_{0,\sigma}(\mathbf{norm}(e'_1), \dots, \mathbf{norm}(e'_k))).$$

Case 4. $e = *_v(e_1)$. We let $e_2 = q_v(e_1)$ and $e_3 = p(e_2)$. By Lemma A.4, $*_v(e_2)$ is equivalent to e , is in $SE(F, V)$ and $h(*_v(e_2)) \leq h(e)$, $|*_v(e_2)| \leq |e|$.

We cannot have e_1 in $F_0 \cup V$ since otherwise e would not have been in $SE(F, V)$. Hence by Lemma A.6, e_3 is defined. Let $e_3 = f(e'_1, \dots, e'_k)$ for some f in F_k , $k \geq 1$. Note that $h(e'_i) < h(e)$ for all $i = 1, \dots, k$. Let e''_i be equal to e'_i if e'_i is minimal and be any equivalent r.e. in $SE(F, V)$ of minimal star-height otherwise.

Once again $h(e''_i) < h(e)$. Hence we can take

$$\mathbf{norm}(e) = *_v(f(\mathbf{norm}(e''_1), \dots, \mathbf{norm}(e''_k))).$$

It is clear that $\mathbf{Val}(\mathbf{norm}(e)) = \mathbf{Val}(e)$, that $h(\mathbf{norm}(e)) \leq h(*_v(e_2)) \leq h(e)$, hence since e is assumed minimal we have the equality and $\mathbf{norm}(e)$ is minimal. Also $\mathbf{norm}(e) \in SE(F, V)$ since $\mathbf{norm}(e''_1), \dots, \mathbf{norm}(e''_k) \in SE(F, V)$; if a rule of (R3) would be applicable at the top level, then e would not be minimal.

Similarly, $\mathbf{norm}(e)$ is strongly minimal since $\mathbf{norm}(e''_1), \dots, \mathbf{norm}(e''_k)$ are so and since if $f(\mathbf{norm}(e''_1), \dots, \mathbf{norm}(e''_k))$ could be replaced by some strictly smaller r.e. in $\mathbf{norm}(e)$, then e would not be minimal.

Finally, $\mathbf{norm}(e)$ is in normal form due to the presence of f after $*_v$ and the inductive hypothesis that $\mathbf{norm}(e''_1), \dots, \mathbf{norm}(e''_k)$ are in normal form too. \square

Proposition A.10 just proved is stronger than Proposition 3.2. in particular due to the fact that the strong minimality is a more severe requirement than Definition 3.1(i).

Let us now turn our attention to the restricted rational expressions and their normal form required by Proposition 5.3.

We shall denote by $SE_0(F, V)$ the set $SE(F, V) \cap E_0(F, V)$.

A.11. Remark. By inspecting the definitions and the proof of the preceding propositions and lemmas, the reader will notice that they all work for r.e.'s in $E_0(F, V)$. More precisely, the operators **simpl**, q_v , p produce expressions in $E_0(F, V)$ from given expressions in $E_0(F, V)$, $SE_0(F, V) - V$ and $SE_p(F, V) \cap E_0(F, V)$, respectively. In the proof of Proposition A.10, the minimal expressions e'_i (in Case 2) and e''_i (in Case 4) are to be taken in $SE_0(F, V)$ of course.

Hence Proposition A.10 holds for restricted rational expressions exactly as for rational expressions.

Proof of Proposition 5.3. (1) Let n be the following operator: $SE_0(F, V) \rightarrow SE_0(F, V)$:

- (i) $n(e) = e$ for e in $F_0 \cup V$.
- (ii) $n(f(e_1, \dots, e_k)) = f(n(e_1), \dots, n(e_k))$.
- (iii) $n(*_v(e)) = *_v(n(q_v(e)))$.

With help of Lemma A.4 it is easy to prove that $n(e)$ is well defined and in $SE_0(F, V)$, that $h(n(e)) \leq h(e)$, and that $Q(n(e)) = Q(e)$ (whence $\mathbf{Val}(n(e)) = \mathbf{Val}(e)$), all this by induction on the structure of e .

And it is clear that $n(e)$ is in normal form (since, in particular, in case (iii) $e \notin F_0 \cup V$, $q_v(e)$ is of the form $f(e_1, \dots, e_k)$ as required).

(2) and (3). It is easy to define a mapping n' from quasi-trees to r.r.e. in $NE_0(F, V)$ such that $n(n'(e)) = e$, and to show that there is a unique such mapping. The results now follow. \square

References

- [1] H. Barendregt, *The Lambda-Calculus*, Studies in Logic **103** (North-Holland, Amsterdam, 1981).
- [2] S. Bloom and C. Elgot, The existence and construction of free iterative theories *J. Comput. System Sci.* **12** (1976) 305–318.
- [3] R. Cohen, Star-height of certain families of regular events, *J. Comput. System Sci.* **4** (1970) 281–297.
- [4] R. Cohen, Techniques for establishing star-height of regular sets, *Math. Systems Theory* **5** (1971) 97–114.
- [5] R. Cohen, Rank non-increasing transformations on transition graphs, *Inform. and Control* **20** (1972) 93–113.
- [6] R. Cohen and J. Brzozowski, General properties of star-height of regular events, *J. Comput. System Sci.* **4** (1970) 260–280.
- [7] B. Courcelle, A representation of trees by languages Part I, *Theoret. Comput. Sci.* **6** (1978) 255–279; Part II, *Theoret. Comput. Sci.* **7** (1978) 25–55.
- [8] B. Courcelle, Fundamental properties of infinite trees, *Theoret. Comput. Sci.* **25** (1983) 95–169.
- [9] G. Cousineau, An algebraic definition for control structures, *Theoret. Comput. Sci.* **12** (1980) 175–192.
- [10] L. Egean, Transition graphs and the star-height of regular events, *Michigan Math. J.* **10** (1963) 385–397.
- [11] S. Eilenberg, *Automata, Languages and Machines, Vol. A* (Academic Press, New York, 1974).
- [12] C. Elgot, Monadic computations and iterative algebraic theories, in: H.E. Rose, ed., *Logic Colloquium '73* (North-Holland, Amsterdam, 1975) pp. 175–230.
- [13] C. Elgot, S. Bloom and R. Tindell, The algebraic structure of rooted trees, *J. Comput. System Sci.* **16** (1978) 362–399.
- [14] S. Ginali, Regular trees and the free iterative theory, *J. Comput. System Sci.* **18** (1979) 228–242.
- [15] G. Huet, Confluent reductions: abstract properties and applications to term rewriting systems, *J. Assoc. Comput. Mach.* **27** (1980) 797–821.
- [16] G. Jacob, Calcul du rang des arbres infinis réguliers, *Proc. CAAP'81*, Lecture Notes Comput. Sci. **112** (Springer, Berlin, 1981) pp. 238–254.
- [17] R. McNaughton, The loop complexity of pure-group events, *Inform. Control* **11** (1967) 167–176.
- [18] R. McNaughton, The loop complexity of regular events, *Inform. Sci.* **1** (1969) 305–328.