Discrete Mathematics and Theoretical Computer Science **2**, 1998, 27–33

Right-cancellability of a family of operations on binary trees

Philippe Duchon

LaBRI, U.R.A. CNRS 1304, Université Bordeaux 1, 33405 Talence, France
E-Mail: `duchon@labri.u-bordeaux.fr`

We prove some new results on a family of operations on binary trees, some of which are similar to addition, multiplication and exponentiation for natural numbers. The main result is that each operation in the family is right-cancellable.

Keywords: binary trees

1 Introduction

The product $a.b$, where a and b are positive integers, can be expressed as the sum of b terms, each being equal to a. Similarly, a^b can be expressed as the product of b factors, each being equal to a. This basically works well because the sum and product operations for integers are associative; to push this process one level further (i.e. define a new operation by iterating exponentiation), one needs to decides on how to order the operations in the expression

$$a \uparrow a \uparrow \ldots \uparrow a$$

(where $\uparrow$ is the exponentiation operation).

One solution is to always perform the operations in a fixed order, usually right-to-left (see Blackley and Borosh [1] or Knuth [2]). Another, richer solution is to use the structure of a binary tree to set the order, and use binary trees instead of integers.

Blondel [3, 4] defines a family of operations on binary trees. Each new operation is defined in terms of the preceding one. The first three operations are generalizations of addition, multiplication and exponentiation for positive integers, while the others have no natural counterpart among positive integers.

The first operation, $\stackrel{1}{\cdot}$, is defined in the following way: if a and b are binary trees, $a \stackrel{1}{\cdot} b$ is the binary tree whose left subtree is a, and whose right subtree is b. Writing trees as parentheses systems, this translates to $a \stackrel{1}{\cdot} b = (ab)$.

Operation $\stackrel{k}{\cdot}$ is defined recursively using $\stackrel{k-1}{\cdot}$:

- $a \stackrel{k}{\cdot} \bullet = a$.
- if $b = (b_L b_R)$, then $a \stackrel{k}{\cdot} b = (a \stackrel{k}{\cdot} b_L) \stackrel{k-1}{\cdot} (a \stackrel{k}{\cdot} b_R)$.

Another way of defining $\overset{k}{\cdot}$ is that the shape of tree b indicates an order in which to compute the $\overset{k-1}{\cdot}$-product of n copies of a, n being the number of leaves of b. For example,

$$a \overset{k}{\cdot} ((\bullet\bullet)(\bullet(\bullet\bullet))) = ((a \overset{k-1}{\cdot} a) \overset{k-1}{\cdot} (a \overset{k-1}{\cdot} (a \overset{k-1}{\cdot} a)))$$

Using the number of leaves as the weight, operations $\overset{1}{\cdot}$, $\overset{2}{\cdot}$ and $\overset{3}{\cdot}$ act as the natural operations of addition, multiplication, and exponentiation, respectively: $|a \overset{1}{\cdot} b| = |a| + |b|$, $|a \overset{2}{\cdot} b| = |a|.|b|$, and $|a \overset{3}{\cdot} b| = |a|^{|b|}$.

Our main result is the proof of a conjecture given by Blondel, which states that all the operations $\overset{k}{\cdot}$ are *right-cancellable*, i.e. for any integer k and trees a, b and c, $a \overset{k}{\cdot} b = c \overset{k}{\cdot} b$ implies $a = c$. This result is easy to prove for $k = 1, 2, 3$; we show that it holds for all $k > 3$.

Sect. 2 introduces a few notations, and recalls some definitions and results on the family of operations. In Sect. 3, the conjecture is reduced to a particular case. The partial ordering defined in Sect. 4 has no direct use in the proof, but appears to be the best to obtain growth results. Sect. 5 redefines the operations using the notion of synthetic attributes, and Sect. 6 finally gives the proof of the conjecture.

2 Notation

The weight (number of leaves) of a tree b will be written $|b|$. When dealing with a word w, $|w|$ will denote its length.

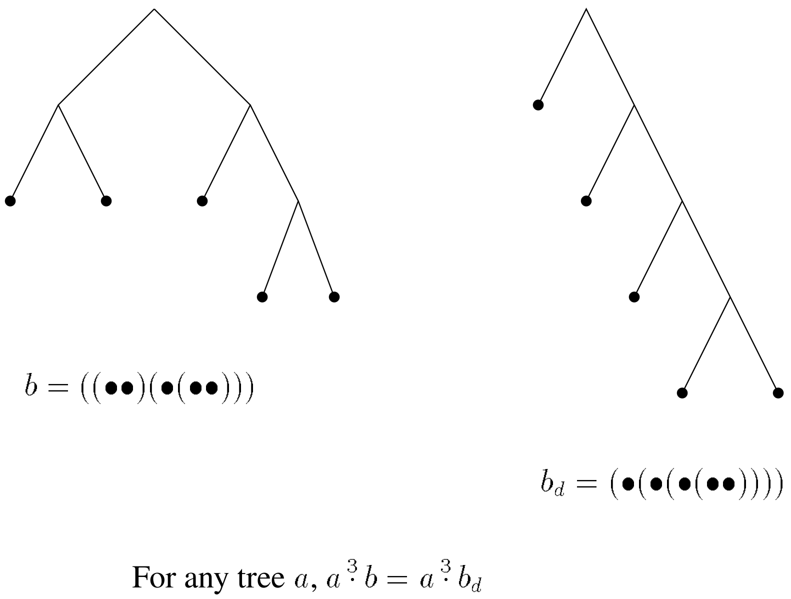

Fig. 1: Two trees with weight 5.

Blondel [3] proves several algebraic properties of the $\overset{k}{\cdot}$ operations, a few of which are recalled below.

- Only operation $\overset{2}{\cdot}$ is associative.

- No $\overset{k}{\cdot}$ is commutative.
- $a \overset{3}{\cdot} b$ depends only on the weight of b, not on its shape, which justifies the use of notation $a \overset{3}{\cdot} |b|$.
- All operations $\overset{k}{\cdot}$ with $k \geq 3$ can be defined in terms of $\overset{3}{\cdot}$ the following way : $a \overset{k}{\cdot} b = a \overset{3}{\cdot} (a \diamond_k b)$, where the natural number $a \diamond_k b$ is defined inductively by:
  - $a \diamond_3 b = |b|$
  - $a \diamond_k \bullet = 1$
  - $a \diamond_k (b_L b_R) = (a \diamond_k b_L) \diamond_{k-1} (a \overset{k}{\cdot} b_R)$ if $k > 3$.
- Each tree has a unique factorization into an ordered $\overset{2}{\cdot}$-product of *prime* binary trees, i.e. trees that cannot be further factorized. Any tree with a prime number of leaves is clearly prime, but the reverse is not true.

Since operation $\overset{2}{\cdot}$ acts somewhat like multiplication, we will write $a.b$ for $a \overset{2}{\cdot} b$. Similarly, since $a \overset{3}{\cdot} b$ does not depend upon the shape of tree b but only on its weight, and is only the result of $a.a \ldots a$ (with $|b|$ factors), we will write $a^{|b|}$ for $a \overset{3}{\cdot} b$.

Using these notations, we have the familiar property $a^{n+m} = a^n.a^m$ (this translates into $a \overset{3}{\cdot} (b \overset{1}{\cdot} c) = (a \overset{3}{\cdot} b) \overset{2}{\cdot} (a \overset{3}{\cdot} c)$, which follows from the definitions of $\overset{3}{\cdot}$ and $\overset{1}{\cdot}$).

3 Preliminary Lemma

To prove the conjecture, we will first reduce it to a simpler form using the following lemmas:

Lemma 1 *Let a, b, c and d be four binary trees, with $a \neq \bullet$ and $c \neq \bullet$, and let k and k' be integers no smaller than 3.*

If $a \overset{k}{\cdot} b = c \overset{k'}{\cdot} d$, then there is a binary tree u and two integers m and n such that $a = u^n$ and $c = u^m$.

Proof. We can rewrite $a \overset{k}{\cdot} b = a^{a \diamond_k b}$ and $c \overset{k'}{\cdot} d = c^{c \diamond_{k'} d}$, so we only need to prove the lemma for $k = k' = 3$.

Consider the factorization of a and c into products of prime trees, and write them as words A and C on the (infinite) alphabet of prime trees. The factorizations of a^r and c^s are, respectively, A repeated r times, and C repeated s times (written A^r and C^s, but bear in mind that $|A^r| = r|A|$). $|A|$, here, is the number of (not necessarily distinct) prime trees involved in the factorization of a.

Since $a^r = b^s$, the same applies to the words: $A^r = C^s$. Let g be the gcd of $|A|$ and $|C|$, and U the left factor of length g of both A and C. Since A, repeated r times, is the same as C, repeated s times, it follows that $A = U^{|A|/g}$, and $C = U^{|C|/g}$.

Converting U back into a binary tree, we get exactly $a = u^n$ and $b = u^m$ with $n = |A|/g$ and $m = |C|/g$. □

Using this lemma, we can reduce the conjecture to the case where trees a and c are powers of a common tree u, which means we only need to prove that, when n and m are different integers, $(u^n) \overset{k}{\cdot} b$ and $(u^m) \overset{k}{\cdot} b$ are different. In fact, when $n < m$, $(u^m) \overset{k}{\cdot} b$ is a vastly larger tree (in most sensible senses of 'larger', including the one defined in the next section) than $(u^n) \overset{k}{\cdot} b$, but we will only prove that $(u^n) \overset{k}{\cdot} b$ is a strict subtree of $(u^m) \overset{k}{\cdot} b$.

4 Partial Ordering of Binary Trees

Operation $\overset{2}{\cdot}$ can be described geometrically in the following way : $a \overset{2}{\cdot} b$ is obtained by changing every leaf of b into a copy of a. Thus, since $u^{n+1} = u.u^n$, we can see that successive powers of a single tree u are prefixes of each other, in the sense that a copy of u^n with the same root is included in u^{n+1}. This property is not limited to powers of a single tree (the same relationship exists between b and $a.b$), but we will only consider such a situation when defining a partial ordering on the set of binary trees.

Definition 2 *Two binary trees a and b are called comparable if there exist a tree u and two integers n and m, such that $a = u^n$ and $b = u^m$. We will then write $a \leq b$ if $n \leq m$.*

The above definition is correct: if more than one tree u can be chosen, n and m will always be in the same order whatever the chosen tree. The defined relation is a partial ordering on the set of binary trees : since $|u^n| = |u|^n$, the relation is the weight semi-ordering, restricted to pairs of trees that are powers of a common tree.

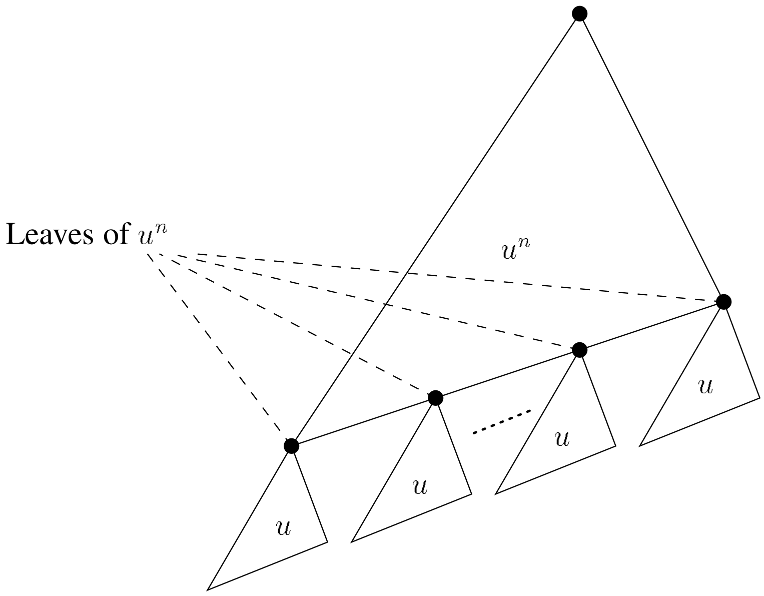

Fig. 2: The identity $u^{n+1} = u.u^n$.

Using this partial ordering, minimal elements are those trees that are not a power of some other tree. Each minimal element is comparable only to its powers, and the order is exactly that of the exponents. Finally, each tree is comparable to exactly one minimal element.

This partial ordering is not explicitly used in the proof of the conjecture. It is only given here because it is the underlying ordering that is somewhat compatible with the $\overset{k}{\cdot}$ operations and related functions (in the sense that $a \leq b$ implies $(a \overset{k}{\cdot} c) \leq (b \overset{k}{\cdot} c$ and $(c \overset{k}{\cdot} a) \leq (c \overset{k}{\cdot} b)$, when $k \geq 3$). Obtaining similar properties for more 'natural' orderings (like those obtained by considering prefix trees, or prime factorizations as subwords or factors of each other) has proved to be difficult.

Definition 3 *Let $k \geq 3$ be an integer, and u a binary tree. We define $f_{u,k}$ as the function of two integer variables n and m defined by*

$$f_{u,k}(n,m) = (u^n) \diamond_k (u^m)$$

or, equivalently,

$$(u^n) \overset{k}{\cdot} (u^m) = (u^n)^{f_{u,k}(n,m)} = u^{n \cdot f_{u,k}(n,m)}$$

When $k = 3$, $f_{u,3}$ is simple: $f_{u,3}(n,m) = |u^m| = |u|^m$. This function is thus strictly increasing according to m (as long as $u \neq \bullet$), and increasing (in fact, constant) according to n. The proof of the conjecture is based on proving that, for each $k > 3$, $f_{u,k}$ is strictly increasing according to both its variables. This translates to the following: operations $\diamond_k$ and $\overset{k}{\cdot}$ are compatible with the ordering defined above as long as their operands are comparable.

Surprisingly enough, having a be a prefix of b is not sufficient, as can be seen by taking $a = ((\bullet(\bullet\bullet))\bullet)$ and $b = ((\bullet(\bullet(\bullet\bullet)))\bullet)$: a is a prefix of b, but $a \overset{3}{\cdot} (\bullet\bullet)$ is not a prefix of $b \overset{3}{\cdot} (\bullet\bullet)^\dagger$.

5 Redefining Operations $\overset{k}{\cdot}$

We now show that operations $\overset{k}{\cdot}$ and the related $f_{u,k}$ functions can be defined in terms of a very particular case of synthetic attributes. Attributes are normally associated to a context-free grammar (see Knuth [5] for a detailed definition), which is a formal rewriting system used to recursively define the structure of the combinatorial objects studied. In the case of binary trees, the simplest thing to do is to say that a binary tree is either the single node $\bullet$, either composed of left and right subtrees which themselves are binary trees. This translates into the formal grammar $T = \bullet + (T.T)$, which is the underlying grammar in all the attributes defined below.

A synthetic attribute can be defined on a binary tree by choosing a two-variable function f (the 'computing rule') and a value to be given to each leaf in the tree (f should be defined on $E \times E$ with values in E, where E is some domain including all values given to leaves). This allows us to compute a value (attribute) for each node in the tree, using the following recurrence rule: if the left and right sons, respectively, of an internal node, have values α and β, then the node has value $f(\alpha, \beta)$. The attribute for the tree is the value of its root node.

Using this context, the definition for $a \overset{k}{\cdot} b$ can be translated into a synthetic attribute computed on binary tree b:

- each leaf in b has value a;
- the computing rule is $\overset{k-1}{\cdot}$: if the left and right sons of an internal node have respective values u and v, this node has value $u \overset{k-1}{\cdot} v$.

This description of $\overset{k}{\cdot}$ corresponds to the fact that, when computing $a \overset{k}{\cdot} b$, the shape of tree b indicates exactly how to associate terms in the calculus of $a \overset{k-1}{\cdot} a \cdots a \overset{k-1}{\cdot} a$ (where a appears $|b|$ times), since this expression is ambiguous whenever $k \neq 3$. For example, if b is the tree in Figure 3 ($b = ((\bullet\bullet)\bullet)$), $a \overset{k}{\cdot} b$ is $(a \overset{k-1}{\cdot} a) \overset{k-1}{\cdot} a$.

When computing $f_{u,k}(n,m)$, we get : $f_{u,k}(n,m)$ is the synthetic attribute value for tree u^m when leaves have value $|u|^n$ and the computing rule is $f_{u,k-1}$.

$\dagger$ There is a 'left, right, right, left, right, right' path from the root in $a \overset{3}{\cdot} (\bullet\bullet)$, but not in $b \overset{3}{\cdot} (\bullet\bullet)$

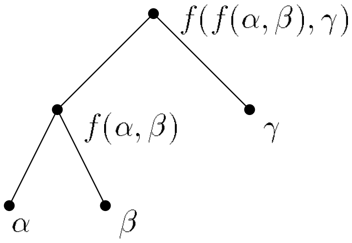

Fig. 3: An example of synthetic attribute.

Proposition 4 (Computing $f_{u,k}(n,m+1)$ on u^m) *$f_{u,k}(n,m+1)$ can be computed as an attribute on u^m (instead of u^{m+1}), still using computing rule $f_{u,k-1}$, by giving leaves a value of $f_{u,k}(n,1)$ instead of $|u|^n$.*

Proof. Recall that u^{m+1} is obtained by replacing all leaves of u^m by copies of u, so while computing $f_{u,k}(n,m+1)$ as an attribute on u^{m+1}, all internal nodes that are leaves in the prefix tree u^m (the roots of copies of u) have value $f_{u,k}(n,1)$. Thus, the value of the root node is not changed when these nodes are considered as leaves with value $f_{u,k}(n,1)$. □

Given a binary tree of weight n and a computing rule f, we can now define a function of n variables as follows: $F(x_1,\ldots,x_n)$ is the attribute computed with rule f on tree a if the leaves (in prefix or symmetric order) have respective values $x_1,\ldots,x_n$. We will use two very simple results:

- **Initial values growth property:** assume the computing rule f is weakly increasing with respect to both its variables; then the resulting function F is also weakly increasing with respect to all of its variables. If f is additionally strictly increasing with respect to its first variable, then F is strictly increasing with respect to its first variable. If f is strictly increasing with respect to all its variables, then so is F.

- **Tree branches growth property:** if f is weakly increasing with respect to one of its variables and strictly increasing with respect to the other, and if for some integer m we have $f(m,m)>m$, then $F(x_1,\ldots,x_n)>\min(x_1,\ldots,x_n)$ provided the minimum is at least m (which is always true if the domain for f is restricted to pairs of positive integers).

These results are easily proved by induction on the height of tree a.

We are now ready to state and prove the main property:

Proposition 5 *Set $k>3$ an integer, and u a binary tree, $u\neq\bullet$. Then the $f_{u,k}$ function is strictly increasing with respect to each of its variables.*

Proof. The proof is by induction on k.

- Assume $k=4$, and recall that $f_{u,4}(n,m)$ is obtained by computing a synthetic attribute on tree u^m with computing rule $f_{u,3}$ and leaf values all set to $|u|^n$. Now $f_{u,3}(n,m)=|u|^m$, so this function is strictly increasing with respect to variable m (and constant with respect to variable n). By the initial values growth property, we can deduce that $f_{u,4}$ is strictly increasing with respect to variable n (if n increases, all leaf values increase).

Now recall that $f_{u,4}(n, m+1)$ can also be obtained by computing the synthetic attribute on tree u^m, with leaf values $f_{u,4}(n, 1)$. Using the tree branches growth property on the computation for $f_{u,4}(n, 1)$ (which uses tree u and leaf values $|u|^n$), we have $f_{u,4}(n, 1) > |u|^n$, which in turns implies (by the initial values growth property) that $f_{u,4}(n, m+1) > f_{u,4}(n, m)$. This proves that $f_{u,4}$ is strictly increasing with respect to both its variables.

- Now set $k > 4$ such that the stated property holds for $k-1$. Replacing $f_{u,4}$ and $f_{u,3}$ by $f_{u,k}$ and $f_{u,k-1}$, respectively, in the above proof, we prove that $f_{u,k}$ is itself strictly increasing with respect to both its variables, thus ending the proof.

□

6 Proof of the Conjecture

We will now prove the following:

Theorem 6 (Right-cancellation) *Set $k > 3$ an integer, and a, b, c three binary trees. If $a \overset{k}{\cdot} b = c \overset{k}{\cdot} b$, then $a = c$.*

Proof. We have already shown that we only need prove this theorem when a and c are powers of some common tree u, i.e. if $(u^n) \overset{k}{\cdot} b = (u^m) \overset{k}{\cdot} b$, then $n = m$ (this is only true if $u \neq \bullet$, but the case when $u = \bullet$ reduces to $a = c = \bullet$ anyway).

We will in fact prove that $n \mapsto (u^n) \diamond_k b$ is strictly increasing. Recall that $(u^n) \diamond_k b$ can be computed as a synthetic attribute on tree b, using $f_{u,k-1}$ as the computing rule and $|u|^n$ as leaf value. Now, we know from Proposition 5 that $f_{u,k-1}$ is strictly increasing with respect to both its variables, which is enough to prove (thanks to the initial values growth property) that $(u^n) \diamond_k b$ increases strictly with n.

Now since $|(u^n) \overset{k}{\cdot} b| = |u|^{n \cdot ((u^n) \diamond_k b)}$, this in turns implies that $|(u^n) \overset{k}{\cdot} b|$ increases strictly with n, thus $(u^n) \overset{k}{\cdot} b$ and $(u^m) \overset{k}{\cdot} b$ can only be equal if $n = m$. □

References

[1] Blakley, G. R. and Borosh, I. (1979). Knuth's iterated powers. *Adv. in Math.*, **34**: 109–136.

[2] Knuth, D. E. (1976). Mathematics and computer science: coping with finiteness. *Science*, **194**: 1235–1242.

[3] Blondel, V. (to appear). Properties of a hierarchy of operations on binary trees. *Acta Informatica*.

[4] Blondel, V. (1995). Une famille d'opérations sur les arbres binaires. *C. R. Acad. Sci. Paris, Série 1*, **321**: 491–494.

[5] Knuth, D. E. (1968). Semantics of context-free languages. *Math. Systems Theory*, **2**: 127–145.