Information Processing Letters 73 (2000) 87–92

Information Processing Letters

www.elsevier.com/locate/ipl

An efficient upper bound of the rotation distance of binary trees

Jean Pallo [1]

LE2I, Université de Bourgogne, B.P. 47870, F21078 Dijon-Cedex, France

Received 11 November 1999; received in revised form 5 January 2000
Communicated by L. Boasson

Abstract

A polynomial time algorithm is developed for computing an upper bound for the rotation distance of binary trees and equivalently for the diagonal-flip distance of convex polygons triangulations. Ordinal tools are used. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Binary trees; Rotation; Triangulation; Lattice; Algorithms

1. Introduction

There exists a well-known one-to-one correspondence between binary trees and convex polygon triangulations. A rotation in a binary tree is a local restructuring of the tree that changes the position of an internal node and one of its children while the symmetric order in the tree is preserved [2,4]. A diagonal-flip transformation is an operation that converts one triangulation of a convex polygon into another by removing a diagonal in the triangulation and adding the diagonal that subdivides the resulting quadrilateral in the opposite way [17]. A system that is isomorphic to binary trees related by rotations is that of triangulations of a convex polygon related by the diagonal-flip transformation. The rotation (respectively diagonal-flip) distance between a pair of binary trees (respectively triangulations of a polygon) is the minimum number of rotations (respectively diagonal-flip transformations) needed to convert one tree (respectively triangulation) into the other. Thus rotation distance of binary trees and diagonal-flip distance of triangulations are equivalent. An open question is the complexity status of computing the rotation distance $d(T, T')$ between two trees T and T' or equivalently the diagonal-flip distance between two triangulations. It is not known whether the problem is NP-complete [16]. On the other hand, the maximum rotation distance between any pair of binary trees with n internal nodes is at most $2n - 6$ for $n \geqslant 11$ [2,9,10,17].

Some authors have worked on triangulations rather than binary trees [1,3,5,7,8,17]. Indeed they think that the diagonal-flip transformation is more natural and intuitive. In this paper, we take the alternative point of view. We start from the fact that rotation induces a lattice structure on binary trees. In Section 2, we recall some known properties of the rotation lattice of binary trees. In Section 3, we define a new operation f on binary trees, called flexion, based on the usual rotation which acts on polygons. If $f(T)$ is the flexed tree corresponding to T, then we have $d(f(T), f(T')) = d(T, T')$. We deduce in Section 4 an efficient algorithm for computing an upper bound $\mu(T, T')$ of $d(T, T')$.

[1] Email: pallo@u-bourgogne.fr.

PII: S0020-0190(00)00008-9

2. Preliminaries

Let denote by $\bigcirc$ (respectively $\square$) external nodes (respectively leaves) of a binary tree. The weight $|T|$ of a binary tree T is the number of leaves of T. Let B_n denote the set of binary trees with n internal nodes (and thus with $n+1$ leaves). The leaves of $T \in B_n$ are numbered from 1 to $n+1$ by a preorder traversal of T. Let T_L (respectively T_R) denote the left (respectively right) subtree of T. Thus we have $T = \bigcirc T_L T_R$ in Polish notation. The mirror image $\widetilde{T}$ of a binary tree T is recursively defined by

$$\widetilde{T} = \bigcirc \widetilde{T_R}\widetilde{T_L} \quad \text{and} \quad \widetilde{\square} = \square.$$

Given $T \in B_n$, the weight sequence of T is the integer sequence $w_T = (w_T(1), \ldots, w_T(n))$ where $w_T(i)$ is the weight of the largest subtree of T whose last leaf is the ith leaf. The left-rotation $\to$ on B_n is defined as follows. A tree $T \in B_n$ being given, it associates a tree T' obtained by replacing some subtree $\bigcirc T_1 \bigcirc T_2 T_3$ of T by the subtree $\bigcirc\bigcirc T_1 T_2 T_3$. Let $\overset{-1}{\to}$ denote the right-rotation and let $\overset{*}{\to}$ denote the reflexive transitive closure of $\to$. The rotation distance $d(T, T')$ between T and $T' \in B_n$ is the minimum number of left- and right-rotations required to convert T into T'.

Given $T, T' \in B_n$, we have $T \overset{*}{\to} T'$ iff for all $i \in [1, n]$:

$$w_T(i) \leqslant w_{T'}(i)$$

[12]. This result allows to prove that $(B_n, \overset{*}{\to})$ is a lattice [12,13]. The weight sequence of the meet of T and $T' \in B_n$ is easy to compute:

$$w_{T \wedge T'}(i) = \min\bigl(w_T(i), w_{T'}(i)\bigr)$$

for all i. The join is built as follows:

for $i := 1$ **to** n **do**
  $w_{T\vee T'}(i) = \max(w_T(i), w_{T'}(i))$
  if $w_{T\vee T'}(i) \neq 1$ **and** $w_{T\vee T'}(i) \neq i$ **then**
    $j := \min\{k - w_{T\vee T'}(k) + 1,$
        $k \in [i - w_{T\vee T'}(i) + 1, i]\}$
    $w_{T\vee T'}(i) := i - j + 1$
  endif
endfor

Given $T, T' \in B_n$ such that $T \overset{*}{\to} T'$, we write $c(T, T')$ for the minimum of the lengths of the chains in $[T, T']$.

The following algorithm computes $c(T, T')$. At each step, it makes use of the first left-rotation which can be applied when traversing the tree in preorder.

function $c(T, T')$
$c := 0$
while $i = \min\{k \mid w_T(k) < w_{T'}(k)\}$ exists **do**
  $j := i - w_T(i) + 1$; $k := j - w_T(j - 1)$;
  $p := \max\{m \geqslant 1 \mid j = m - w_T(m) + 1\}$
  $w_T(p) := p - k + 1$; $c := c + 1$
endwhile

If $T, T' \in B_n$, computing $w_{T\vee T'}$ and $c(T, T')$ requires $O(n^2)$ time complexity in the worst case since $c \leqslant 2n$. An upper bound $\delta(T, T')$ for $d(T, T')$ is obtained by the following formula [13] which computes the lengths of two different paths between T and T', namely $(T, T \wedge T', T')$ and $(T, T \vee T', T')$:

$$\begin{aligned} d(T, T') \leqslant \min\bigl\{& c(T \wedge T', T) + c(T \wedge T', T'), \\ & c(T, T \vee T') + c(T', T \vee T')\bigr\} \\ = {}& \delta(T, T'). \end{aligned}$$

Computing $\delta(T, T')$ runs in $O(n^2)$ time in the worst case.

3. Flexion

Let us consider $(n+2)$-gons, i.e., convex polygons with $n+2$ sides and with a distinguished side as the top. We label the other sides from 1 to $n+1$ counterclockwise. Any triangulation of the $(n+2)$-gon has n triangles and $n-1$ non-crossing diagonals. There is an explicit bijection between B_n and the triangulations of $(n+2)$-gons. The top of the $(n+2)$-gon corresponds to the root of the tree. The ith side of the $(n+2)$-gon corresponds to the ith leaf of the tree. Diagonals correspond to internal nodes recursively as shown in Fig. 1. See also Fig. 2.

We denote by $b(T)$ the $(n+2)$-gon corresponding to $T \in B_n$.

Definition. Flexion is a transformation f on B_n defined as follows: $T' = f(T)$ if $b(T')$ is obtained from $b(T)$ by rotating $b(T)$ counterclockwise by $360/(n+2)$ degrees.

Clearly we have $f^{n+2}(T) = T$. Now, we want to compute $w_{f(T)}$ from w_T. If $T_R = \square$, we have

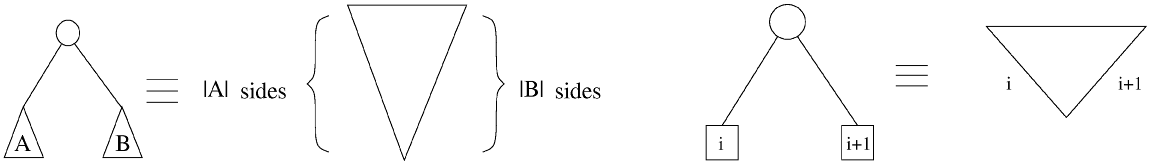

Fig. 1.

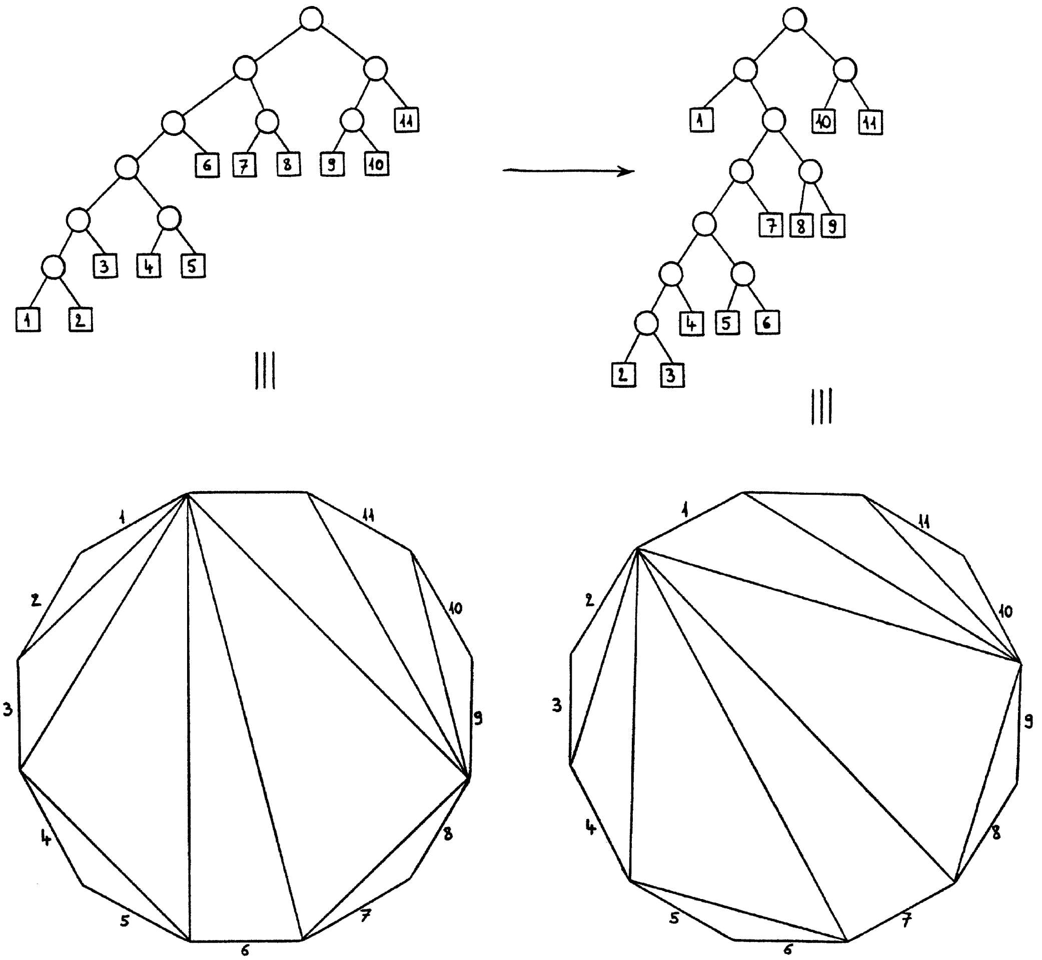

Fig. 2.

$f(T)_L = \square$ and $f(T)_R = T_L$. If $T_R \neq \square$, $f(T)$ depends on the subtrees on the right arm of T. If $T = \bigcirc T_L \bigcirc B_1 \bigcirc B_2 \bigcirc \ldots \bigcirc B_q \square$ with $q \geqslant 1$, then $f(T) = \bigcirc\bigcirc \ldots \bigcirc \square T_L B_1 \ldots B_{q-1} B_q$.

Given $T \in B_n$, we need to compute the number l of the last leaf of the left subtree T_L and the number k of the first leaf of the subtree B_q which is the left brother of the last leaf of T. Given the weight sequence w of $T \in B_n$, the following algorithm computes the weight sequence w' of $f(T)$ (a loop **for** $i := j$ **to** k is not executed if $k < j$):

```
flexion(w, w', n)
k := n − w(n) + 1;
l := max{i ∈ [1, n] | w(i) = i};
w'(1) := 1;
for j := 2 to l do
  w'(j) := w(j − 1)
endfor
if l = n then exit endif
if l = n − 1 then
  w'(n) = n; exit endif;
  m := l + 1;
while m < k − 1 do
  p := max{i ∈ [m, k − 1] | w(i) = i − m + 1};
  for j := m to p − 1 do
    w'(j + 1) := w(j)
  endfor
  w'(p + 1) := p + 1; m := p + 1;
endwhile; w'(k) := k;
for j := k + 1 to n do
  w'(j) := w(j − 1)
endfor
endflexion
```

The work done by *flexion* is proportional to the number of comparisons required to compute l and p, checking whether $w(i) = i$ and $w(i) = i - m + 1$ for the current i. In the best case where $T_R = \square$ (thus with $k = 1$ and $l = n$), the complexity is O(1). In the worst case where $w = (1, 1, \ldots, 1)$, the complexity is $O(n^2)$. The number of times p is computed in the **while** loop is $\nu - 1$ where ν is the number of internal nodes on the right arm of T. ν is the level sequence of the last leaf of T. It has been proved in [15] and cited in [14] that the average level number of the last leaf of a binary tree is asymptotically equal to 3. Therefore the average asymptotic complexity is $O(n)$. See an example in Fig. 2 with $w_T = (1, 2, 3, 1, 5, 6, 1, 8, 1, 2)$ and $w_{f(T)} = (1, 1, 2, 3, 1, 5, 6, 1, 9, 1)$.

Let us conclude this section by a combinatorial remark. Defining two trees $T, T' \in B_n$ as equivalent if $b(T)$ can be obtained from $b(T')$ by a rotation, one can see the equivalence classes are enumerated by the sequence **M3288** $= (1, 4, 6, 19, 49, 150, 442, \ldots)$ of [18]. If we add to the equivalence condition the fact that $\widetilde{T} = T'$ (i.e., $b(T)$ and $b(T')$ coincide under reflection), we get a new equivalence relation for which equivalence classes are now enumerated by the sequence **M2375** $= (1, 3, 4, 12, 27, 82, 228, \ldots)$ of [18]. These two sequences have appeared in scattered places throughout the literature [6,11].

4. Computing an upper bound

The following theorem holds for the flexion transformation f defined in the previous section:

Theorem. *$d(f(T), f(T')) = d(T, T')$ for all $T, T' \in B_n$.*

Proof. Rotation distance of binary trees and diagonal-flip distance of triangulations of convex polygons are equivalent, i.e., $d(T, T') = d(b(T), b(T'))$ where b is the bijection between binary trees and polygons defined in Section 3. By definition, $d(b(T), b(T'))$ is the minimum number of diagonal-flip transformations needed to convert $b(T)$ into $b(T')$. Let us consider the flexed trees $f(T)$, $f(T')$, and the corresponding polygons $b(f(T))$, $b(f(T'))$ obtained respectively from $b(T)$ and $b(T')$ by a rotation of $+360/(n+2)$ degrees. Then converting $b(f(T))$ into $b(f(T'))$ requires the same minimum number of diagonal-flip transformations than converting $b(T)$ into $b(T')$, i.e., $d(b(f(T)), b(f(T'))) = d(b(T), b(T'))$. Thus $d(f(T), f(T')) = d(T, T')$. $\square$

Corollary. *If $T, T' \in B_n$, an upper bound $\mu(T, T')$ of the rotation distance $d(T, T')$ is given by*:

$$\mu(T, T') = \min_{0 \leqslant i \leqslant n+1} \delta\big(f^i(T), f^i(T')\big).$$

Proof. In Section 2, we have exhibited an upper bound $\delta(T, T')$ of the rotation distance $d(T, T')$ using ordinal tools. According to the above theorem, an other upper bound is given by $\delta(f(T), f(T'))$. Using the flexion transformation, other upper bounds are obtained since $f^{n+2}(T) = T$ and $f^{n+2}(T') = T'$. $\square$

The following remark is critical. Consider $T, T' \in B_n$ such that $w_T = (1, 1, \ldots, 1, 2)$ and $w_{T'} = (1, 1, \ldots, 1, n-1, n)$. Then we have $\delta(T, T') = c(T, T') = 2n - 4$ whereas $d(T, T') = n$. Now consider the corresponding flexed trees $f(T)$ and $f(T')$. We have:

$$w_{f(T)} = (1, 2, 3, \ldots, n-2, n-1, 1),$$
$$w_{f(T')} = (1, 1, \ldots, 1, 1, n-1),$$

	τ		τ'	$\delta(\tau,\tau')$
T:	1 1 1 1 2 3 4 6 1 1 10 1 2 14	T':	1 2 1 2 1 2 3 4 1 1 1 10 1 1	19
$f(T)$:	1 1 1 1 1 2 3 4 6 1 1 10 1 2	$f(T')$:	1 1 3 1 2 1 2 3 4 1 1 1 13 14	18
$f^2(T)$:	1 2 3 1 1 1 2 3 4 6 1 1 13 1	$f^2(T')$:	1 1 1 3 1 2 1 2 3 4 1 1 1 13	18
$f^3(T)$:	1 1 2 3 1 1 1 2 3 4 6 1 1 14	$f^3(T')$:	1 2 1 1 3 1 2 1 2 3 4 1 1 1	17
$f^4(T)$:	1 1 1 2 3 1 1 1 2 3 4 6 1 1	$f^4(T')$:	1 1 3 1 1 6 1 8 1 2 3 12 13 14	15
$f^5(T)$:	1 2 3 1 2 6 7 1 1 2 3 4 13 14	$f^5(T')$:	1 1 1 3 1 1 6 1 8 1 2 3 12 13	15
$f^6(T)$:	1 1 2 3 1 2 6 7 1 1 2 3 4 13	$f^6(T')$:	1 2 1 1 3 1 1 6 1 8 1 2 3 12	15
$f^7(T)$:	1 2 1 2 3 1 2 6 7 1 1 2 3 4	$f^7(T')$:	1 1 3 1 1 3 1 1 6 1 8 1 2 3	16
$f^8(T)$:	1 1 3 1 2 3 1 2 6 10 11 1 2 3	$f^8(T')$:	1 1 1 4 1 1 3 1 1 6 1 12 1 2	20
$f^9(T)$:	1 1 1 3 1 2 3 1 2 6 10 12 1 2	$f^9(T')$:	1 1 1 1 4 1 1 3 1 1 6 1 13 1	19
$f^{10}(T)$:	1 1 1 1 3 1 2 3 1 2 6 10 13 1	$f^{10}(T')$:	1 1 1 1 1 4 1 1 3 1 1 6 1 14	18
$f^{11}(T)$:	1 1 1 1 1 3 1 2 3 1 2 6 10 14	$f^{11}(T')$:	1 1 1 1 1 1 4 1 1 3 1 1 6 1	18
$f^{12}(T)$:	1 1 1 1 1 1 3 1 2 3 1 2 6 10	$f^{12}(T')$:	1 2 3 4 1 1 1 8 1 1 3 1 1 14	17
$f^{13}(T)$:	1 2 3 4 5 1 1 3 1 2 3 1 2 6	$f^{13}(T')$:	1 1 2 3 4 1 1 1 8 1 1 3 1 1	17
$f^{14}(T)$:	1 1 2 3 4 6 1 1 9 1 2 3 1 2	$f^{14}(T')$:	1 2 1 2 3 4 1 1 1 10 1 1 13 14	18
$f^{15}(T)$:	1 1 1 2 3 4 6 1 1 10 1 2 13 1	$f^{15}(T')$:	1 1 2 1 2 3 4 1 1 1 10 1 1 13	19
$f^{16}(T)$:	1 1 1 1 2 3 4 6 1 1 10 1 2 14	$f^{16}(T')$:	1 2 1 2 1 2 3 4 1 1 1 10 1 1	19

Fig. 3.

$$w_{f(T)\wedge f(T')} = (1,1,\ldots,1,1),$$
$$w_{f(T)\vee f(T')} = (1,2,3,\ldots,n-1,n),$$

and

$$\delta\big(f(T), f(T')\big) = \min\big((n-2)+(n-2), 1+(n-1)\big) = n$$

which is exactly $d(T,T')$. The above remark underlines the benefit of this method. The flexion transformation allows to find n as approximation of $d(T,T')$ instead of $2n-4$.

Computing $\mu(T,T')$ requires *flexion* $2n$ times, *function c* $4n$ times, *join* and *meet* n times. Therefore it runs in $O(n^3)$ time in the worst case. The space complexity is clearly $O(n)$. The example in Fig. 3 illustrates how the algorithm works for two trees T and $T' \in B_{14}$. In this case $\mu(T,T') = 15$ which is exactly equal to $d(T,T')$.

5. Conclusion

For T and $T' \in B_n$, it will be of interest to compute the average ratio $r = \mu(T,T')/d(T,T')$ as a function of n, which appears to be hard. However, computer experiments underline the fact that $\mu(T,T')$ is equal to $d(T,T')$ most of the time. More precisely, if we randomly generate a lot of pairs of trees in B_{14}, we obtain as average ratio $r = 1.015$.

References

[1] D. Aldous, Triangulating the circle, at random, Amer. Math. Monthly 101 (1994) 223–233.
[2] K. Culik, D. Wood, A note on some tree similarity measures, Inform. Process. Lett. 15 (1982) 39–42.
[3] R.D. Dutton, R.O. Rogers, Properties of the rotation graph of binary trees, Congr. Numer. 109 (1995) 51–63.
[4] C. Germain, J. Pallo, The number of coverings in four Catalan lattices, Internat. J. Comput. Math. 61 (1996) 19–28.
[5] S. Hanke, T. Ottmann, S. Schuierer, The edge-flipping distance of triangulations, J. Universal Comput. Sci. 2 (1996) 570–579.

[6] F. Harary, E.M. Palmer, R.C. Read, On the cell-growth problem for arbitrary polygons, Discrete Math. 11 (1975) 371–389.
[7] F. Hurtado, M. Noy, J. Urrutia, Flipping edges in triangulations, Discrete Comput. Geom. 22 (1999) 333–346.
[8] M. Li, L. Zhang, Better approximation of diagonal-flip transformation and rotation transformation, in: W.L. Hsu (Ed.), Proc. 4th Annual Conf. Computing and Combinatorics, Lecture Notes in Computer Science, Vol. 1449, Springer, Berlin, 1998, pp. 85–94.
[9] F. Luccio, L. Pagli, On the upper bound on the rotation distance of binary trees, Inform. Process. Lett. 31 (1989) 57–60.
[10] E. Mäkinen, On the rotation distance of binary trees, Inform. Process. Lett. 26 (1987/88) 271–272.
[11] J.W. Moon, L. Moser, Triangular dissections of n-gons, Canad. Math. Bull. 6 (1963) 175–178.
[12] J. Pallo, Enumerating, ranking and unranking binary trees, Comput. J. 29 (1986) 171–175.
[13] J. Pallo, On the rotation distance in the lattice of binary trees, Inform. Process. Lett. 25 (1987) 369–373.
[14] F. Ruskey, On the average shape of binary trees, SIAM J. Alg. Disc. Meth. 1 (1980) 43–50.
[15] F. Ruskey, T.C. Hu, Generating binary trees lexicographically, SIAM J. Comput. 6 (1977) 745–758.
[16] D.D. Sleator, R.E. Tarjan, W.R. Thurston, Rotation distance, in: T.M. Cover, B. Gopinath (Eds.), Open Problems in Communication and Computation, Springer, Berlin, 1987, pp. 130–137.
[17] D.D. Sleator, R.E. Tarjan, W.R. Thurston, Rotation distance, triangulations and hyperbolic geometry, J. Amer. Math. Soc. 1 (1988) 647–681.
[18] N.J.A. Sloane, S. Plouffe, The Encyclopedia of Integer Sequences, Academic Press, New York, 1995.