• 16000 11463
còpia 1

**Average-case Analysis
of Equality of Binary Trees
Under the BST Probability Model**

Conrado Martínez

Report LSI-91-8

Average-case Analysis of Equality of Binary Trees Under the BST Probability Model*

C. Martínez

Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Pau Gargallo 5, 08028-Barcelona
e-mail : conrado@lsi.upc.es

March 14, 1991

Abstract

In this paper a simple algorithm to test equality of binary trees currently used in symbolic computation, unification, etc. is investigated. Surprisingly enough, it takes $O(1)$ steps on average to decide if a given pair of trees of total size n are equal or not if the uniform probability model for the input is assumed. Moreover, other similar algorithms have qualitatively the same average complexity behavior. In this paper, we analyze this average complexity when the so-called bst probability model is assumed. The analysis is itself more complex although feasible, involving partial differential equations and singularity analysis of Bessel functions. Nevertheless, partial differential equations are generally unsolvable, like the one which is derived from the bivariate recurrences for the equality, and an indirect mechanism solving a simpler equation and showing asymptotical equivalence of solutions is used to obtain the main result : testing equality of a pair of binary trees of total size n and distributed accordingly to the bst probability model is $\Theta(\log n)$ on the average.

1 Introduction

The uniform probability model for the input is a usual assumption in many average-case analysis due to several reasons. There are, for example, many applications where this model accurately enough approximates the actual probability distribution of the input. Moreover, the assumption has been historically

---

*This research was partially supported by the ESPRIT BRA Program of the EC under contract no. 3075, project ALCOM.

used to cope with uncertainty situations such as the one that arises if the actual distribution is unknown. On the other hand, because of the limitations of our techniques [Kar86, Sed83] only in a few cases we are able to carry on the analysis for other probabilistic models than the simplest one: the uniform probability model. Finally, even if the results based on the uniform model might not always be considered full relevant, they may be of large value for comparing the performance of different algorithms which solve the given problem.

Nevertheless, it would be interesting to do these analysis for other reasonable probability models and compare the differences, if any, existing between the average complexities derived for different models. Moreover, it would be interesting to compare “discriminative” power of these models; how different are the relative comparisons of the complexities of distinct algorithms. The objective of such an analysis is to elucidate the sensitivity issues of the various models.

In this paper we examine the average behavior of the equality test for binary trees assuming the probability model for binary search trees (BST, for short) or bst probability model. As we shall see in Section 2, for this non-uniform model well balanced trees have high probability whereas poorly balanced trees have low probability.

Equality test of binary trees is an important algorithm in its own, but also it is the core of many other algorithms for binary trees, for example simplification [CS86] and it is closely related to other algorithms such as tree matching [SF83] and unification [CDS89]. It turns out that the equality test algorithm has $O(1)$ average complexity under the uniform distribution of the input: no matter what is the size of the given trees it takes a constant number of steps on the average to decide if they are equal or not.

In the previous work [BCDM91] the intersection of two binary trees has been considered assuming that they were distributed according to the model to be defined in Section 2. It has been shown there that the intersection of binary trees, as defined in [CKS89], has a sublinear average complexity under the new considered input model, far from the constant behavior obtained when the input model is the uniform one.

This work shows that testing equality is no longer of constant cost for the bst probability model, but of logarithmic cost. Our results clearly contrast with those obtained using the uniform model, which predicts qualitatively identical average behaviors for equality test and the intersection algorithms (constant). They are of different orders of magnitude if the bst probability model is assumed instead.

We also show how techniques similar to those used in the analysis of intersection can be applied for our problem: mainly the method of Riemann for hyperbolic partial differential equations and singularity analysis for the estimation of coefficients of the Bessel functions involved.

In Section 2 the bst probability model for trees is defined as well as its extension to pairs of trees. Section 3 is devoted to the formulation of the problem

and it is also shown that the problem can be reduced to that of solving a partial differential equation. The difficulty to solve it leads us to seek for bounds of the average complexity time of the equality test algorithm. Therefore, we start getting an upper bound in Section 3. In Section 4 our efforts are directed towards the obtainment of a lower bound. We introduce and analyze a related algorithm which yields a logarithmic lower bound.

In Section 5 we deal again with the initial problem and it is shown that the average complexity of the equality test is of the same order of magnitude as the lower bound.

2 The Probabilistic Model for Binary Trees

In the sequel we denote the left subtree and the right subtree of a tree T by T^l and T^r , respectively. Moreover, the set of binary trees will be denoted $\mathcal{B}$ and the set of binary trees of size n by $\mathcal{B}_n$.

Recall that a *binary search tree* is a node-labeled binary tree whose nodes are labeled in increasing order in the inorder traversal of the tree. Binary search trees have the following recursive definition [VF90]: Given a sequence of n keys $S = (k_1, k_2, \dots, k_n)$ from a totally ordered set, the binary search tree of S is defined recursively as follows:

$$BST(S) = \begin{cases} < BST(S_l), k_1, BST(S_g) > & \text{if } |S| \geq 1 \\ < \square > & \text{otherwise,} \end{cases} \quad (2.1)$$

where $|S|$ denotes the number of keys in S and S_l (S_g) denotes the subsequence of S formed by the elements of S which are smaller (larger) than k_1 , and $\square$ denotes the empty binary tree.

In the model of probability associated with binary search trees, each sequence S is obtained by consecutively sampling at random n elements from a real interval, or equivalently, as far as the relative ordering is concerned, the elements form a random permutation of size n .

Therefore, each permutation has probability $1/n!$, and if we denote by $p(T)$ the probability of tree T , we have the following definition (implicitly appearing in [Knu73, exercise 5, 6.2.2])

Definition 2.1 *The bst probability model for binary trees is recursively defined by*

$$p(T) = \begin{cases} 1 & \text{if } T = \square \\ \frac{p(T^l) \cdot p(T^r)}{1 + |T^l| + |T^r|} & \text{otherwise.} \end{cases}$$

The recursive manner in which we express this probability distribution is very handy to simplify some proofs about the average behaviour of binary search.

Note that the probability model induced for the BSTs generated from these equally likely permutations is not uniform (i.e. trees of the same size may not have the same probability).

The problem we are going to present involves pairs of binary search trees, so we must extend the probability model to pairs of trees. We are confronted with two situations :

If each of the trees is drawn *independently* of the other, the probability of the pair is just

$$p_{\text{ind}}(T_1, T_2) = p(T_1) \cdot p(T_2).$$

Notice that in this case for all n and m ,

$$\sum_{|T_1|=n, |T_2|=m} p_{\text{ind}}(T_1, T_2) = 1.$$

On the other hand, if we consider pairs of binary trees (T_1, T_2) such that $|T_1| + |T_2| = n$, i.e. n is the total size of the input, the probability of the pair is

$$p(T_1, T_2) = \frac{p(T_1) \cdot p(T_2)}{1 + |T_1| + |T_2|},$$

which corresponds to the model in which all the $n + 1$ possible partitions of n into $n = i + j$ (being $|T_1| = i$ and $|T_2| = j$) are equally likely.

Notice that this probability coincides with the probability of a tree formed by a root and the subtrees T_1 and T_2 .

In this case we have that for all $n \geq 0$,

$$\sum_{|T_1|+|T_2|=n} p(T_1, T_2) = 1.$$

3 Testing Equality of Binary Trees

The investigated algorithm tests whether two given trees are equal or not.

In order to test the equality of two binary trees, we proceed as follows : if one of the given trees is a leaf we determine with a unique comparison if the other is also a leaf or not; if both are not leaves, first (and recursively) test if their left subtrees are equal; if so, continue testing equality of the right subtrees and then return saying true or false depending on the last test, otherwise, return false (see Figure 1).

If we denote by $e(T_1, T_2)$ the number of steps that this algorithm performs on input (T_1, T_2) then

$$e(T_1, T_2) = \begin{cases} 1 & \text{if } \square \in \{T_1, T_2\} \\ 1 + e(T_1^l, T_2^l) & \text{if } \square \notin \{T_1, T_2\} \text{ and } T_1^l \neq T_2^l \\ 1 + e(T_1^l, T_2^l) + e(T_1^r, T_2^r) & \text{otherwise} \end{cases}$$

We wish to compute the average value of $e(T_1, T_2)$ over all pairs (T_1, T_2) of joint size $|T_1| + |T_2| = n$. Let E_n be this average value; then

---

```

function Equal?( $T_1, T_2$  : BinTree) returns Boolean is
1      if  $\square \in \{T_1, T_2\}$  then
2          return ( $T_1 = T_2$ )
3      else
4          if Equal?( $T_1^l, T_2^l$ ) then
5              return Equal?( $T_1^r, T_2^r$ )
6          else
7              return false
8          endif
9      endif
endfunction;

```

Figure 1: Equality Test Algorithm

$$E_n = \sum_{|T_1|+|T_2|=n} p(T_1, T_2) \cdot e(T_1, T_2) \quad (3.1)$$

Following the standard technique of generating functions [VF90, GJ83] we define the following generating functions,

$$E(z) = \sum_{T_1, T_2 \in \mathcal{B}} p(T_1, T_2) e(T_1, T_2) z^{|T_1|+|T_2|} \quad (3.2)$$

and

$$E(x, y) = \sum_{T_1, T_2 \in \mathcal{B}} p(T_1) p(T_2) e(T_1, T_2) x^{|T_1|} y^{|T_2|} \quad (3.3)$$

where $E_n = [z^n]E(z)$ (E_n is the n^{th} coefficient of the Taylor expansion of $E(z)$) and the two generating functions are related by,

$$E(z) = \frac{1}{z} \int_0^z E(t, t) dt \quad (3.4)$$

Using the decomposition of $\mathcal{B} \times \mathcal{B}$ suggested by the recursive definition of $e(T_1, T_2)$, that is,

$$\begin{aligned}
 \mathcal{B} \times \mathcal{B} &= (\square, \square) + \square \times (\mathcal{B} - \square) + (\mathcal{B} - \square) \times \square + \\
 &+ \{(T_1, T_2) \mid T_1^l \neq T_2^l\} + \\
 &+ \{(T_1, T_2) \mid T_1^l = T_2^l\}
 \end{aligned} \quad (3.5)$$

and Definition 2.1 we obtain the following hyperbolic partial differential equation ([Cop75]) for $E(x, y)$,

$$\frac{\partial^2 E}{\partial x \partial y} = \frac{1}{(1-x)^2(1-y)^2} + E(x, y) \left(\frac{1}{(1-x)(1-y)} + Q(x, y) \right) \quad (3.6)$$

where $Q(x, y) = \sum_{T \in \mathcal{B}} p^2(T) x^{|T|} y^{|T|}$ and the initial conditions are,

$$\forall x \quad E(x, 0) = \frac{1}{1-x} \quad (3.7)$$

$$\forall y \quad E(0, y) = \frac{1}{1-y} \quad (3.8)$$

The generating function $Q(x, y)$ depends, in fact, on $x \cdot y$. We can define therefore,

$$R(z) = \sum_{T \in \mathcal{B}} p^2(T) z^{|T|} \quad (3.9)$$

so $Q(x, y) = R(x \cdot y)$. $R(z)$ satisfies the following second order ordinary differential equation,

$$z \frac{d^2 R}{dz^2} + \frac{dR}{dz} - R^2(z) = 0 \quad (3.10)$$

with $R(0) = R'(0) = 1$.

Equation (3.10) seems very difficult to be solved analytically, mainly because its non-linearity. Moreover, $R(z)$ has a finite movable singularity which depends on the initial conditions [Inc56]. This makes unfeasible the derivation of an exact solution for Eq. (3.6).

In order to obtain the answer for the average complexity of the equality, we will first devise an upper and a lower bound for it, and then prove that this complexity actually reaches the lower bound.

A crude upper bound is easily obtained supposing that in the previous algorithm, we replace the conditional execution of the equality test over right subtrees (lines 4–8) by

return **Equal?** (T_1^l, T_2^l) **and** **Equal?** (T_1^r, T_2^r)

This version works correctly but in an inefficient way, since it tests the equality of the right subtrees even when left subtrees are not equal. Denoting by $d(T_1, T_2)$ the number of steps that this modified algorithm makes given input (T_1, T_2) we obviously have, for each pair $(T_1, T_2) \in \mathcal{B} \times \mathcal{B}$, $e(T_1, T_2) \leq d(T_1, T_2)$. Let D_n be the average complexity of this dully version of the equality test. Of course, for each $n \geq 0$, $E_n \leq D_n$. It is easy to verify that D_n coincides with the average complexity of the algorithm of intersection of binary trees (shuffle) and then $E_n = O(n^K / \sqrt{\log n})$, $K = 2(\sqrt{2} - 1)$ under the assumption that input is distributed according to the bst probability model [BCDM91].

Recall that both tests (that defined in Figure 1 and the dully proposed version) have the same worst-case complexity and constant average complexity when the input model is the uniform one.

```

function BackboneEqual?( $T_1, T_2 : \text{BinTree}$ ) returns Boolean is
    if  $\square \in \{T_1, T_2\}$  then
        return ( $T_1 = T_2$ )
    else
        return BackboneEqual?( $T_1^l, T_2^l$ )
    endif
endfunction;

```

Figure 2: Backbones Equality Test

---

4 A Tight Lower Bound for Equality

Being Eq. (3.10) seemingly very difficult to be solved, is there any chance to get an estimate for E_n ? An upper bound has been considered in Section 3, and it is possible to get a lower bound as well.

For instance, an algorithm that tests if the leftmost branches, also called *backbones*, of two given trees are equal or not would take no more steps than to test equality between the same pair of trees (see Figure 2).

The number of steps $l(T_1, T_2)$ it takes to decide if T_1 and T_2 have the same backbone satisfies $l(T_1, T_2) \leq e(T_1, T_2)$, for all pairs (T_1, T_2) , and similarly, the average number of steps done by algorithm **BackboneEqual?** with input of size n , say L_n , verifies $L_n \leq E_n$.

Proceeding as in Section 3, let us define,

$$L(z) = \sum_{T_1, T_2 \in \mathcal{B}} l(T_1, T_2) p(T_1, T_2) z^{|T_1|+|T_2|} \quad (4.1)$$

and

$$L(x, y) = \sum_{T_1, T_2 \in \mathcal{B}} l(T_1, T_2) p(T_1) p(T_2) x^{|T_1|} y^{|T_2|} \quad (4.2)$$

Eqs. (4.1,4.2) bear the same relation as Eqs. (3.2,3.3),

$$L(z) = \frac{1}{z} \int_0^z L(t, t) dt \quad (4.3)$$

Moreover, let $L_n = [z^n]L(z)$.

The bivariate generating function $L(x, y)$ is characterized by the following partial differential equation,

$$\frac{\partial^2 L}{\partial x \partial y} = \frac{1}{(1-x)^2(1-y)^2} + \frac{L(x, y)}{(1-x)(1-y)} \quad (4.4)$$

subjected to the initial conditions,

$$\forall x \quad L(x, 0) = \frac{1}{1-x} \quad (4.5)$$

$$\forall y \quad L(0, y) = \frac{1}{1-y} \quad (4.6)$$

A particular solution of Eq. (4.4) is easily obtained, so we can express $L(x, y)$ as,

$$L(x, y) = L_h(x, y) + \frac{1}{2} \frac{\ln\left(\frac{1}{1-x}\right) + \ln\left(\frac{1}{1-y}\right)}{(1-x)(1-y)} \quad (4.7)$$

where $L_h(x, y)$ satisfies the homogeneous equation,

$$\frac{\partial^2 L_h}{\partial x \partial y} = \frac{L_h(x, y)}{(1-x)(1-y)} \quad (4.8)$$

and the initial conditions are,

$$\forall x \quad L_h(x, 0) = \frac{1}{1-x} \left(1 - \frac{1}{2} \ln\left(\frac{1}{1-x}\right)\right) \quad (4.9)$$

$$\forall y \quad L_h(0, y) = \frac{1}{1-y} \left(1 - \frac{1}{2} \ln\left(\frac{1}{1-y}\right)\right) \quad (4.10)$$

Recall that we are interested in obtaining the asymptotic behavior of $[z^n]L(z)$; by Eq. (4.7), $[z^n]L(z) = \frac{1}{n+1}[z^n]L(z, z)$, which gives,

$$\begin{aligned} L_n = [z^n]L(z) &= \frac{1}{n+1} \left([z^n]L_h(z, z) + [z^n] \frac{\ln\left(\frac{1}{1-z}\right)}{(1-z)^2} \right) = \\ &= \frac{1}{n+1} [z^n]L_h(z, z) + H_{n+1} - 1 \end{aligned} \quad (4.11)$$

where H_n is the n^{th} harmonic number.

Applying the method of Riemann [Cop75, ch. 5], [CH62, ch. 5, pp. 449–461], [BCDM91] to Eq. (4.8), and imposing Eqs. (4.9, 4.10) we obtain a solution in terms of Bessel functions. Some further calculations lead to the following technical lemma, left without proof, which establishes that

Lemma 4.1

$$[z^n]L_h(z, z) \approx 3[z^n]J_0\left(2i \ln\left(\frac{1}{1-z}\right)\right)$$

where $J_0(x)$ is the zeroth order Bessel function of the first kind.

Using the Laplace method for integrals [DB58, ch.4] for the integral representation of $J_0(x)$ or the standard asymptotic expansion [AS64] of $J_0(x)$, for $|x| \rightarrow \infty$ together with the appropriate transfer lemmas yields,

$$[z^n]J_0\left(2i\ln\left(\frac{1}{1-z}\right)\right) \approx \frac{1}{\sqrt{\pi}}\frac{n}{\sqrt{\ln n}}(1+O(\frac{1}{\ln n})) \quad (4.12)$$

Hence, this last result together with Eq. (4.11) and Lemma 4.1 gives us

$$\begin{aligned} L_n &\approx H_{n+1} - 1 + \frac{1}{\sqrt{\pi \ln n}} + O(\ln^{-3/2} n) \\ &\approx \ln n + O(1) \end{aligned} \quad (4.13)$$

remembering that $H_n \approx \ln n + \gamma + O(1/n)$.

We point out that if we assume the uniform distribution of the input the average complexity of this algorithm is also $O(1)$; this fact shows that the bst probability model predicts qualitative differences between the complexity of algorithms where the former does not.

5 The Average Complexity of Equality Testing

The question that now arises is how far is E_n from the bounds obtained :

$$\begin{aligned} E_n &= \Omega(\log n) \\ E_n &= O\left(\frac{n^K}{\sqrt{\log n}}\right), \quad K = 2(\sqrt{2} - 1) \end{aligned}$$

Is it of the same order of magnitude as any of them?

If we consider the length of the backbone of binary trees distributed according to the bst probability model, it is not difficult to prove that its expectation is $O(\log n)$, whereas its standard deviation is $O(\sqrt{\log n})$. So the probability that two trees of size n have the same backbone is small : $O(1/\sqrt{\log n})$. This fact strongly suggests that $E_n = O(L_n)$.

A closer look at Eqs. (3.6,4.4) reveals that they are very similar, being the unique difference the term $Q(x, y)$ in the multiplicative factor of the unknown function. Moreover, $E(x, y)$ and $L(x, y)$ have only Taylor series development in the region $\{(x, y) \mid |x| < 1, |y| < 1\}$, while $Q(x, y)$'s convergence domain is larger and thus, it can be considered "neglectable".

This observation leaded us to consider whether there was a relationship between the solutions of both equations, and more precisely how were related their respective Riemman-Green functions (see [Cop75],[CH62]).

These functions are the solutions of the homogeneous equations,

$$\frac{\partial^2 \hat{V}}{\partial x \partial y} = \hat{V} \left(\frac{1}{(1-x)(1-y)} + R(x \cdot y) \right) \quad (5.1)$$

$$\frac{\partial^2 V}{\partial x \partial y} = \frac{V}{(1-x)(1-y)} \quad (5.2)$$

where both functions share the same initial conditions :

$$\begin{aligned}
\left. \frac{\partial V}{\partial x} \right|_{y=0} &= \left. \frac{\partial \hat{V}}{\partial x} \right|_{y=0} = 0 \\
\left. \frac{\partial V}{\partial y} \right|_{x=0} &= \left. \frac{\partial \hat{V}}{\partial y} \right|_{x=0} = 0 \\
V(0,0;0,0) &= \hat{V}(0,0;0,0) = 1
\end{aligned} \tag{5.3}$$

And

$$\begin{aligned}
E(x, y) &= E(0,0)\hat{V}(0,0;x,y) + \\
&+ \int_0^y \hat{V}(0,t;x,y) \left. \frac{\partial E}{\partial y} \right|_{y=t} dt + \int_0^x \hat{V}(t,0;x,y) \left. \frac{\partial E}{\partial x} \right|_{x=t} dt + \\
&+ \int_0^x \int_0^y \hat{V}(\xi,\nu;x,y) \frac{1}{(1-\xi)^2(1-\nu)^2} d\xi d\nu
\end{aligned} \tag{5.4}$$

$$\begin{aligned}
L(x, y) &= L(0,0)V(0,0;x,y) + \\
&+ \int_0^y V(0,t;x,y) \left. \frac{\partial L}{\partial y} \right|_{y=t} dt + \int_0^x V(t,0;x,y) \left. \frac{\partial L}{\partial x} \right|_{x=t} dt + \\
&+ \int_0^x \int_0^y V(\xi,\nu;x,y) \frac{1}{(1-\xi)^2(1-\nu)^2} d\xi d\nu
\end{aligned} \tag{5.5}$$

If we make a change of variables

$$\begin{cases} X &= \ln \left(\frac{1}{1-x} \right) \\ Y &= \ln \left(\frac{1}{1-y} \right) \end{cases}$$

the corresponding Riemman-Green functions are characterized by simpler equations and thus, the following lemma, binding $V(X_0, Y_0; X, Y)$ and $\hat{V}(X_0, Y_0; X, Y)$, can be established

Lemma 5.1

$$\hat{V}(X_0, Y_0; X, Y) \sim V(X_0, Y_0; X, Y)$$

when $X \rightarrow \infty, Y \rightarrow \infty$.

Undoing the change of variables, and applying the above lemma to Eqs. (5.4,5.5) we have

$$E(x, y) \sim L(x, y) \quad x \rightarrow 1, y \rightarrow 1$$

Hence, the n^{th} coefficient of $E(z)$ behaves asymptotically as L_n ,

$$[z^n]E(z) = \frac{1}{n+1}[z^n]E(z, z) \approx \frac{1}{n+1}[z^n]L(z, z) = L_n$$

And therefore, it's immediate to state our main theorem,

Theorem 5.1 *Under the bst probability model, the average complexity time of the equality test for binary trees behaves as*

$$E_n = \Theta(\log n) \tag{5.6}$$

6 Conclusions. Further Research

It seems that the kind of mathematical problems here emerging rely on the nature of the assumed probability model rather than on the problem. Therefore, Riemman's method for hyperbolic partial differential equations as well as singularity analysis of the ubiquitous Bessel functions will be helpful for other analyses based on the bst probability model. Unfortunately, the partial differential equations corresponding to the recurrences for the average complexities to be calculated, will be almost never explicitly solvable and it will be interesting to get ready mathematical tools which reduce difficult problems to simpler ones. This work can be considered as a first step towards this direction, since a mathematical technique (equivalence of Riemman-Green functions) is the fundamental bedrock for our results.

Further research will concentrate in this line trying to develop other more powerful techniques to carry on these analyses.

7 Acknowledgements

I thank R. Casas and J. Díaz for encouragement and advice. I also wish to thank P. Flajolet, J. Grané, J. Gruska and X. Messeguer for many interesting and useful comments and suggestions.

References

- [AS64] M. Abramowitz and I.A. Stegun, editors. *Handbook of Mathematical Functions*. Dover Publications, Inc., New York, 1964.
- [BCDM91] R. Baeza-Yates, R. Casas, J. Díaz, and C. Martínez. On the average size of the intersection of binary trees. *SIAM J. of Computing*, to be published.
- [CDS89] R. Casas, J. Díaz, and J. M. Steyaert. Average-case analysis of Robinson's unification algorithm with two different variables. *Information Processing Letters*, 31:227–232, June 1989.

- [CH62] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 2. InterScience J. Wiley, 1962.
- [CKS89] C. Choppy, S. Kaplan, and M. Soria. Complexity analysis of term-rewriting systems. *Theoretical Computer Science*, 67(2):261–282, 1989.
- [Cop75] E.T. Copson. *Partial Differential Equations*. Cambridge University Press, Cambridge, 1975.
- [CS86] R. Casas and J. M. Steyaert. Bottom-up recursion in trees. In *Proc. CAAP’86, LNCS 214*, pages 172–182. Springer-Verlag, 1986.
- [DB58] N.G. De Bruijn. *Asymptotic Methods in Analysis*. Dover Publ., New York, 1958.
- [GJ83] I. Goulden and D. Jackson. *Combinatorial Enumerations*. Wiley, New York, 1983.
- [Inc56] E. L. Ince. *Ordinary Differential Equations*. Dover Publ., New York, 1956.
- [Kar86] R. Karp. Combinatorics, complexity and randomness. *Comm. of the ACM*, 29(2):98–119, Feb 1986.
- [Knu73] D.E. Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison-Wesley, Reading, Mass., 1973.
- [Sed83] R. Sedgewick. *Mathematical Analysis of Combinatorial Algorithms*, pages 124–205. Academic Press, 1983.
- [SF83] J. M. Steyaert and P. Flajolet. Patterns and pattern-matching in trees: an analysis. *Information and Control*, 58(1-3):19–58, 1983.
- [VF90] J.S. Vitter and P. Flajolet. *Average-case Analysis of Algorithms and Data Structures*, volume 1, chapter 9. North-Holland, 1990.

Figure 3: Riemann's solution domain

Appendix A

Riemann's method was devised to obtain solutions of second-order linear partial differential equations (PDEs). Therefore it can be applied to equations of the form

$$L[u] = au_{xx} + 2hu_{xy} + bu_{yy} + 2gu_x + 2fu_y + cu = F(x, y)$$

where $u = u(x, y)$, subscripts denote partial differentiation with respect to the indicated variable(s) as usual and a, b, h, g, f and c are functions of x and y alone. From now on, we shall assume $a = b = f = g = 0$, since this is the case with the PDEs we are interested in.

Let the characteristic axes be $x = x_0$ and $y = y_0$, for which Cauchy data is given, that is, u, u_x and u_y are known, and let $D = (x_0, y_0)$.

Let $P = (\xi, \eta)$, $Q = (x_0, \eta)$, $R = (\xi, y_0)$, and let γ be the domain bounded by PQ, PR and C (see Figure 3).

Using Green's theorem it can be shown that

$$\begin{aligned} u(\xi, \eta) &= \frac{1}{2}[u(x_0, \eta)v(x_0, \eta; \xi, \eta) + u(\xi, y_0)v(\xi, y_0; \xi, \eta)] - \\ &\quad - \frac{1}{2} \int_D^Q \frac{\partial v}{\partial y} u(x_0, y) dy + \frac{1}{2} \int_D^Q \frac{\partial u}{\partial y} v(x_0, y; \xi, \eta) dy + \\ &\quad - \frac{1}{2} \int_D^R \frac{\partial v}{\partial x} u(x, y_0) dx + \frac{1}{2} \int_D^R \frac{\partial u}{\partial x} v(x, y_0; \xi, \eta) dx + \\ &\quad + \frac{1}{2} \iint_{\gamma} v(x, y; \xi, \eta) F(x, y) dx dy \end{aligned} \quad (\text{A.1})$$

where $v(x, y; \xi, \eta)$ is the so-called Riemman-Green function and must verify

$$\begin{aligned} L[v] &= 0 \\ v_x &= f v = 0 \quad \text{on } y = y_0 \\ v_y &= g v = 0 \quad \text{on } x = x_0 \\ \text{and} \quad v(x_0, y_0; x_0, y_0) &= 1 \end{aligned} \quad (\text{A.2})$$

In order to obtain Riemman-Green functions, we shall try a Hadamard series formula,

$$v(x, y; \xi, \eta) = \sum_{n \geq 0} v_n \frac{\Gamma^n}{n!^2} \quad (\text{A.3})$$

where v_n are functions to be determined and $\Gamma = (x - \xi)(y - \eta)$.

It can be shown that

$$\begin{aligned} v_0 &= 1 \\ v_n &= -\frac{n v_0}{r^n} \int_0^r \frac{s^{n-1}}{v_0} L[v_{n-1}] ds \end{aligned} \quad (\text{A.4})$$

where $x = \xi + r \sin \theta$, $y = \eta + r \cos \theta$, and the linear differential operator L is applied with respect to $\alpha = \xi + s \sin \theta$ and $\beta = \eta + s \cos \theta$ instead of x and y .

Making the change of variables presented in Section 5

$$\begin{cases} X &= \ln \left(\frac{1}{1-x} \right) \\ Y &= \ln \left(\frac{1}{1-y} \right) \end{cases}$$

then Eqs. (5.1,5.2) are transformed to

$$\frac{\partial^2 \hat{V}}{\partial X \partial Y} = \hat{V} \left[1 + R((1 - e^{-X}) \cdot (1 - e^{-Y})) e^{-(X+Y)} \right] \quad (\text{A.5})$$

$$\frac{\partial^2 V}{\partial X \partial Y} = V \quad (\text{A.6})$$

Imposing conditions (A.2) to Eq. (A.6) it turns out that the coefficients of the Hadamard series development of V are $V_n = 1$, $n \geq 0$, so we obtain

$$V(X, Y; \xi, \eta) = J_0 \left(2i \sqrt{(X - \xi)(Y - \eta)} \right)$$

On the other hand, if Eq. (A.4) is used to determine the coefficients $\hat{V}_n$ in the Hadamard series development of $\hat{V}$ we can show that

$$\begin{aligned} \hat{V}_0 &= 1 \\ \hat{V}_n &= -\frac{n}{r^n} \left[\int_0^r s^{n-1} \left(\frac{\partial^2 \hat{V}_{n-1}}{\partial \alpha \partial \beta} - \hat{V}_{n-1} \right) ds - \right. \\ &\quad \left. - \int_0^r s^{n-1} \hat{V}_{n-1} h(\alpha, \beta) ds \right] \end{aligned} \quad (\text{A.7})$$

where $h(x, y) = R((1 - e^{-x})(1 - e^{-y}))e^{-(x+y)}$, $X = \xi + r \sin \theta$, $Y = \eta + r \cos \theta$, $\alpha = \xi + s \sin \theta$ and $\beta = \eta + s \cos \theta$.

Now, we will sketch an induction proof for Lemma 5.1. For $n = 0$, $\hat{V}_0 = V_0 = 1$. Assume that, for all $j < n$, if $X \rightarrow \infty$ and $Y \rightarrow \infty$, we have $\hat{V}_j \sim V_j$. Then, from Eq. (A.7)

$$\hat{V}_n \sim V_n + \frac{n}{r^n} \int_0^r s^{n-1} h(\alpha, \beta) ds \quad (\text{A.8})$$

The following lemma should be stated in order to finish the proof,

Lemma 7.1 *$R(z)$ is analytic in $|z| \leq 1$, with $R(0) = 1$ and $R(1) \leq 1 + \frac{1}{(1-1/\sqrt{6})^2} \approx 3.856 \dots$. Moreover, $R(|z|)$ is a monotonic increasing function for $|z| \geq 0$.*

Its proof is not difficult. First get the recurrence for the coefficients of $R(z)$ given the differential equation it verifies. Once this has been done, a simple inductive proof allows to apply Weierstrass' majoration criterion.

Eq. (A.8) and the previous lemma yield

$$\hat{V}_n \sim V_n + \frac{nK}{a^n r^n} e^{-(\xi+\eta)} \int_0^r t^{n-1} e^{-t} dt \quad (\text{A.9})$$

where K is a constant and $a = \sin \theta + \cos \theta$. Since $X \rightarrow \infty$ and $Y \rightarrow \infty$, it follows that $r \rightarrow \infty$ and the above equation is

$$\hat{V}_n \sim V_n + \frac{n!K}{a^n r^n} e^{-(\xi+\eta)} \sim V_n, \quad r \rightarrow \infty \quad (\text{A.10})$$

Hence, Lemma 5.1 follows.