Contents

Methodological papers

ANNALES MATHEMATICAE ET INFORMATICAE 36. (2009)

ANNALES MATHEMATICAE ET INFORMATICAE

TOMUS 36. (2009)

ESZTERHÁZY KÁROLY COLLEGE
EGER • 1774

HUNGARIA, EGER

ANNALES MATHEMATICAE ET INFORMATICAE

VOLUME 36. (2009)

INSTITUTE OF MATHEMATICS AND INFORMATICS
ESZTERHÁZY KÁROLY COLLEGE
HUNGARY, EGER

HU ISSN 1787-5021 (Print)
HU ISSN 1787-6117 (Online)

A kiadásért felelős:
az Eszterházy Károly Főiskola rektora
Megjelent az EKF Líceum Kiadó gondozásában
Kiadóvezető: Kis-Tóth Lajos
Felelős szerkesztő: Zimányi Árpád
Műszaki szerkesztő: Tómács Tibor
Megjelent: 2009. december   Példányszám: 50

Készítette:
az Eszterházy Károly Főiskola nyomdája
Felelős vezető: Kérészy László

Annales Mathematicae et Informaticae
36 *(2009) pp. 3–14*
http://ami.ektf.hu

A common fixed point theorem via a generalized contractive condition

Abdelkrim Aliouche[a], Faycel Merghadi[b]

[a]Department of Mathematics, University of Larbi Ben M'Hidi Oum-El-Bouaghi, Algeria
[b]Department of Mathematics, University of Tebessa, Algeria

Submitted 8 January 2009; Accepted 12 June 2009

Abstract

We prove a common fixed point theorem for mappings satisfying a generalized contractive condition which generalizes the results of [3, 4, 12, 15, 19, 20, 24] and we correct the errors of [7, 12, 20].

Keywords: Metric space, weakly compatible mappings, common fixed point.

MSC: 47H10, 54H25

1. Introduction

Sessa [21] defined S and T to be weakly commuting as a generalization of commuting if for all $x \in X$.

$$d(STx, TSx) \leqslant d(Tx, Sx).$$

Jungck [9] defined S and T to be compatible as a generalization of weakly commuting if

$$\lim_{n\to\infty} d(STx_n, TSx_n) = 0$$

whenever $\{x_n\}$ is a sequence in X such that $\lim_{n\to\infty} Sx_n = \lim_{n\to\infty} Tx_n = t$ for some $t \in X$. It is easy to show that commuting implies weakly commuting implies compatible and there are examples in the literature verifying that the inclusions are proper, see [9, 21]. Jungck et al [10] defined S and T to be compatible mappings of type (A) if

$$\lim_{n\to\infty} d\left(STx_n, T^2x_n\right) = 0 \quad \text{and} \quad \lim_{n\to\infty} d\left(TSx_n, S^2x_n\right) = 0,$$

whenever $\{x_n\}$ is a sequence in X such that $\lim_{n\to\infty} Sx_n = \lim_{n\to\infty} Tx_n = t$ for some $t \in X$. Example are given to show that the two concepts of compatibility are independent, see [10]. Recently, Pathak and Khan [16] defined S and T to be compatible mappings of type (B) as a generalization of compatible mappings of type (A) if

$$\lim_{n\to\infty} d\left(TSx_n, S^2x_n\right) \leqslant \frac{1}{2}\left[\lim_{n\to\infty} d\left(TSx_n, Tt\right) + \lim_{n\to\infty} d\left(Tt, T^2x_n\right)\right],$$
$$\lim_{n\to\infty} d\left(STx_n, T^2x_n\right) \leqslant \frac{1}{2}\left[\lim_{n\to\infty} d\left(STx_n, St\right) + \lim_{n\to\infty} d\left(St, S^2x_n\right)\right],$$

whenever $\{x_n\}$ is a sequence in X such that $\lim_{n\to\infty} Sx_n = \lim_{n\to\infty} Tx_n = t$ for some $t \in X$. Clearly compatible mappings of type (A) are compatible mappings of type (B), but the converse is not true, see [16]. However, compatible mappings of type (A) and compatibility of type (B) are equivalent if S and T are continuous, see [16]. Pathak et al [17] defined S and T to be compatible mappings of type (P) if

$$\lim_{n\to\infty} d\left(S^2x_n, T^2x_n\right) = 0,$$

whenever $\{x_n\}$ is a sequence in X such that $\lim_{n\to\infty} Sx_n = \lim_{n\to\infty} Tx_n = t$ for some $t \in X$. However, compatibility, compatibility of type (A) and compatibility of type (P) are equivalent if S and T are continuous, see [17]. Pathak et al [18] defined S and T to be compatible mappings of type (C) as a generalization of compatible mappings of type (A) if

$$\lim_{n\to\infty} d\left(TSx_n, S^2x_n\right) \leqslant \frac{1}{3}\left[\lim_{n\to\infty} d\left(TSx_n, Tt\right) + \lim_{n\to\infty} d\left(Tt, S^2x_n\right) + \lim_{n\to\infty} d\left(Tt, T^2x_n\right)\right],$$
$$\lim_{n\to\infty} d\left(STx_n, T^2x_n\right) \leqslant \frac{1}{3}\left[\lim_{n\to\infty} d\left(STx_n, St\right) + \lim_{n\to\infty} d\left(St, T^2x_n\right) + \lim_{n\to\infty} d\left(St, S^2x_n\right)\right],$$

whenever $\{x_n\}$ is a sequence in X such that $\lim_{n\to\infty} Sx_n = \lim_{n\to\infty} Tx_n = t$ for some $t \in X$. Compatibility, compatibility of type (A) and compatibility of type (C) are equivalent if S and T are continuous, see [18]. Pant [15] defined S and T to be reciprocally continuous if

$$\lim_{n\to\infty} STx_n = St \text{ and } \lim_{n\to\infty} TSx_n = Tt,$$

whenever $\{x_n\}$ is a sequence in X such that $\lim_{n\to\infty} Sx_n = \lim_{n\to\infty} Tx_n = t$ for some $t \in X$. It is clear that if S and T are both continuous, then they are reciprocally continuous, but the converse is not true. Moreover, it was proved in [15] that in the setting of common fixed point theorem for compatible mappings satisfying contractive conditions, the continuity of one of the mappings S and T implies their reciprocal continuity, but not conversely.

2. Preliminaries

Definition 2.1 (See [11]). S and T are said to be weakly compatible if they commute at their coincidence points; i.e., if $Su = Tu$ for some $u \in X$, then $STu = TSu$.

Lemma 2.2 (See [9, 10, 16, 17, 18]). *If S and T are compatible, or compatible of type (A), or compatible of type (P), or compatible of type (B), or compatible of type (C), then they are weakly compatible.*

The converse is not true in general, see [4].

Definition 2.3 (See [13]). S and T are said to be $R-$weakly commuting if there exists an $R > 0$ such that

$$d\,(STx, TSx) \leqslant Rd\,(Tx, Sx) \text{ for all } x \in X. \tag{2.1}$$

Definition 2.4 (See [14]). S and T are pointwise $R-$weakly commuting if for all $x \in X$, there exists an $R > 0$ such that (2.1) holds.

It was proved in [14] that R-weakly commutativity is equivalent to commutativity at coincidence points; i.e., S and T are pointwise R-weakly commuting if and only if they are weakly compatible.

Lemma 2.5 (See [22]). *For any $t \in (0, \infty)$, $\psi(t) < t$ iff $\lim_{n\to\infty} \psi^n(t) = 0$, where ψ^n denotes the n-times repeated composition of ψ with itself.*

Several authors proved fixed point and common fixed point theorem for mappings satisfying contractive conditions of integral type, see [1, 3, 4, 5, 6, 7, 12, 19, 20]. The following theorem was proved by [3].

Theorem 2.6 (See [3]). *Let A, B, S and T be self-mappings of a metric space (X, d) satisfying*

$$S\,(X) \subset B\,(X) \quad \textit{and} \quad T(X) \subset A\,(X)\,,$$

$$\int_0^{d(Sx,Ty)} \varphi(t)\,\mathrm{d}t \leqslant \psi\left(\int_0^{M(x,y)} \varphi(t)\,\mathrm{d}t\right)$$

for all $x, y \in X$, $\psi\colon \mathbb{R}_+ \to \mathbb{R}_+$ is a right continuous function such that $\psi(0) = 0$ and $\psi(s) < s$ for all $s > 0$ and $\varphi\colon \mathbb{R}_+ \to \mathbb{R}_+$is a Lebesgue integrable mapping which is summable and satisfies

$$\int_0^{\epsilon} \varphi(t)\,\mathrm{d}t > 0,$$

$$M(x, y) = \max\left\{ d(Ax, By), d(Sx, Ax), d(Ty, By), \frac{1}{2}[d(Sx, By) + d(Ty, Ax)] \right\}.$$

If one of $A\,(X)\,, B\,(X)\,, S(X)$ and $T\,(X)$ is a complete subspace of X, then A and S have a coincidence point and B and T have a coincidence point. Further, if S and A as well as T and B are weakly compatible, then A, B, S and T have a unique common fixed point in X.

Recently, Zhang [24] and Aliouche [2] proved common fixed point theorems using generalized contractive conditions in metric spaces.

Let $A \in (0, \infty]$, $R_A^+ = [0, A)$ and $F\colon R_A^+ \to \mathbb{R}$ satisfying

(i) $F(0) = 0$ and $F(t) > 0$ for each $t \in (0, A)$,
(ii) F is nondecreasing on R_A^+,
(iii) F is continuous.
Define $\mathcal{F}[0, A) = \{F : F$ satisfies (i)–(iii)$\}$.

Lemma 2.7 (See [24]). *Let $A \in (0, \infty]$, $F \in \mathcal{F}[0, A)$. If $\lim_{n\to\infty} F(\epsilon_n) = 0$ for $\epsilon_n \in R_A^+$, then $\lim_{n\to\infty} \epsilon_n = 0$.*

The following examples were given in [24].
(i) Let $F(t) = t$, then $F \in \mathcal{F}[0, A)$ for each $A \in (0, \infty]$.
(ii) Suppose that φ is nonnegative, Lebesgue integrable on $[0, A)$ and satisfies

$$\int_0^\epsilon \varphi(t)\,\mathrm{d}t > 0 \text{ for each } \epsilon \in (0, A).$$

Let $F(t) = \int_0^t \varphi(s)\,\mathrm{d}s$, then $F \in [0, A)$.
(iii) Suppose that ψ is nonnegative, Lebesgue integrable on $[0, A)$ and satisfies

$$\int_0^\epsilon \psi(t)\,\mathrm{d}t > 0 \text{ for each } \epsilon \in (0, A)$$

and φ is nonnegative, Lebesgue integrable on $\left[0, \int_0^A \psi(s)\,\mathrm{d}s\right)$ and satisfies

$$\int_0^\epsilon \varphi(t)\,\mathrm{d}t > 0 \text{ for each } 0 < \epsilon < \int_0^A \psi(s)\,\mathrm{d}s.$$

Let $F(t) = \int_0^{\int_0^t \psi(s)\,\mathrm{d}s} \varphi(u)\,\mathrm{d}u$, then $F \in \mathcal{F}[0, A)$.

(iv) If $G \in [0, A)$ and $F \in \mathcal{F}[0, G(A-0))$, then a composition mapping $F \circ G \in \mathcal{F}[0, A)$. For instance, let $H(t) = \int_0^{F(t)} \varphi(s)\,\mathrm{d}s$, then $H \in \mathcal{F}[0, A)$ whenever $F \in \mathcal{F}[0, A)$ and φ is nonnegative, Lebesgue integrable on $\mathcal{F}[0, F(A-0))$ and satisfies

$$\int_0^\epsilon \varphi(t)\,\mathrm{d}t > 0 \text{ for each } \epsilon \in (0, F(A-0)).$$

Let $A \in (0, \infty]$ and $\psi\colon R_A^+ \to \mathbb{R}_+$ satisfying
(i) $\psi(t) < t$ for all $t \in (0, A)$
(ii) ψ is upper semi-continuous.
(iii) ψ is nondecreasing on R_A^+,
Define $\Psi[0, A) = \{\psi : \psi$ satisfies (i)-(iii)$\}$.

3. Main results

Theorem 3.1. *Let (X, d) be a metric space and $D = \sup\{d(x, y) : x, y \in X\}$. Set $A = D$ if $D = \infty$ and $A > D$ if $D < \infty$. Let A_1, A_2, S and T be self-mappings of (X, d) satisfying*

$$A_1(X) \subset T(X) \text{ and } A_2(X) \subset S(X),$$

$$F(d\,(A_1x, A_2y)) \leqslant \psi(F(L(x, y))) \tag{3.1}$$

for all x, y in X, where

$$L(x,y) = \max\Big\{d(Sx,Ty), d(A_1x,Sx), d(Ty,A_2y), \frac{1}{2}[d(Sx,A_2y) + d(A_1x,Ty)]\Big\},$$

$F \in \digamma[0,A)$ and $\psi \in \Psi[0,F(A-0))$ for all $A \in (0,\infty]$. Suppose that the pair (A_1,S) is weakly compatible and there exists $w \in C(A_2,T)$: the set of coincidence points of A_2 and T such that $A_2Tw = TA_2w$. If one of $A_1\,(X)\,, A_2(X), S\,(X)$ and $T\,(X)$ is a complete subspace of X, then A_1, A_2, S and T have a unique common fixed point in X.

Proof. Let x_0 be arbitrary point in X. Inductively, we can define a sequence $\{y_n\}$ in X such that

$$y_{2n} = A_1x_{2n} = Tx_{2n+1} \text{ and } y_{2n+1} = Sx_{2n+2} = A_2x_{2n+1}$$

for all $n = 0, 1, 2, \ldots$. As in the proof of [2], $\{y_n\}$ is a Cauchy sequence in X. Assume that $S\,(X)$ is complete. Therefore

$$\lim_{n\to\infty} A_1x_{2n} = \lim_{n\to\infty} Tx_{2n+1} = \lim_{n\to\infty} A_2x_{2n+1} = \lim_{n\to\infty} Sx_{2n+1} = z = Su$$

for some $u \in X$. If $A_1u \neq z$ using (3.1) we obtain

$$F(d\,(A_1u, A_2x_{2n+1})) \leqslant \psi(F(L(u, x_{2n})))$$

where

$$L(u,x_{2n}) = \max\Big\{d(Su,Tx_{2n+1}), d(A_1u,Su), d(Tx_{2n+1},A_2x_{2n+1}), \\ \frac{1}{2}[d(Su,A_2x_{2n+1}) + d(A_1u,Tx_{2n+1})]\Big\}.$$

Letting $n \to \infty$, we get

$$F(d\,(A_1u, z)) \leqslant \psi(F(d\,(A_1u, z))) < F(d\,(A_1u, z))$$

which is a contradiction and so $z = A_1u = Su$. If $z \neq A_2w$, applying (3.1) we obtain

$$F(d\,(A_1u, A_2w)) \leqslant \psi(F(d\,(A_1u, A_2w)))$$

where

$$L(u,v) = \max\Big\{d(Su,Tw), d(A_1u,Su), d(Tw,A_2w), \frac{1}{2}[d(Su,A_2w) + d(A_1u,Tw)]\Big\}.$$

Hence

$$F(d\,(z, A_2w)) \leqslant \psi(F(d\,(z, A_2w))) < F(d\,(z, A_2w)).$$

which is a contradiction and so $z = A_1u = Su = A_2w = Tw$.

Since the pairs (A_1, S) is weakly compatible and there exists $w \in C(A_2, T)$ such that $A_2Tw = TA_2w$, we have $Sz = A_1z$ and $Tz = A_2z$.

If $A_1z \neq z$ we have by (3.1)

$$F(d\,(A_1z, A_2w)) \leqslant \psi(F(L(z, w)))$$

where

$$L(z, w) = \max\Big\{d(Sz, Tw), d(A_1z, Sz), d(Bw, A_2w), \frac{1}{2}[d(Sz, A_2w) + d(A_1z, Tw)]\Big\}.$$

Therefore

$$F(d\,(A_1z, z)) \leqslant \psi(F(d\,(A_1z, z))) < F(d\,(A_1z, z))$$

and so $A_1z = Sz = z$. Similarly, we can prove that $A_2z = Tz = z$.

The proof is similar when $T\,(X)$ is assumed to be a complete subspace of X. The case in which $A_1\,(X)$ or $A_2\,(X)$ is a complete subspace of X is similar to the case in which $T\,(X)$ or $S\,(X)$ respectively is complete since $A_1\,(X) \subset T\,(X)$ and $A_2\,(X) \subset S\,(X)$. The uniqueness of z follows from (3.1). □

Theorem 3.1 generalizes Theorem 2.6 of [3].

Corollary 3.2. *Let (X, d) be a metric space and $D = \sup\{d(x, y) : x, y \in X\}$. Set $A = D$ if $D = \infty$ and $A > D$ if $D < \infty$. Let $\{A_i\}$, $i = 1, 2, \dots, S$ and T be self-mappings of (X, d) satisfying*

$$A_1(X) \subset T(X) \text{ and } A_i(X) \subset S(X),\ i \geqslant 2$$

and

$$F(d\,(A_1x, A_iy)) \leqslant \psi(F(L_i(x, y))),\ i \geqslant 2$$

for all x, y in X, where

$$L_i(x, y) = \max\Big\{d(Sx, Ty), d(A_1x, Sx), d(A_iy, Ty), \frac{1}{2}[d(Sx, A_iy) + d(A_1x, Ty)]\Big\},$$

$F \in \digamma[0, A)$ and $\psi \in \Psi[0, F(A - 0))$ for all $A \in (0, \infty]$. Suppose that the pair (A_1, S) is weakly compatible and there exists $w \in C(A_i, T)$: the set of coincidence points of A_i and T such that $A_iTw = TA_iw$ for some $i \geqslant 2$. If one of $A_i\,(X)$, $S\,(X)$ and $T\,(X)$ is a complete subspace of X. Then A_i, S and T have a unique common fixed point in X.

If $\varphi(t) = 1$ in Corollary 3.2, we get a generalization of a theorem of [15]. The following example illustrates our corollary 3.2.

Example 3.3. Let $X = [0, 10]$ be endowed with the metric $d\,(x, y) = |x - y|$,

$$Sx = \begin{cases} 0, & \text{if } x = 0, \\ x + 8, & \text{if } x \in (0, 2]\,, \\ x - 2, & \text{if } x \in (2, 10]\,, \end{cases} \qquad Tx = \begin{cases} 0, & \text{if } x = 0, \\ x + 5, & \text{if } x \in (0, 2]\,, \\ x - 2, & \text{if } x \in (2, 10]\,, \end{cases}$$

$$A_1x=\begin{cases}3, & \text{if } x\in(0,2]\,,\\ 0, & \text{if } x\in\{0\}\cup(2,10]\,,\end{cases}\qquad A_2x=\begin{cases}0, & \text{if } x\in[0,2]\,,\\ 4, & \text{if } x\in(2,10]\,,\end{cases}$$

$$A_3x=\begin{cases}0, & \text{if } x\in[0,2]\,,\\ 5, & \text{if } x\in(2,10]\,,\end{cases}\qquad A_4x=\begin{cases}0, & \text{if } x\in[0,2]\,,\\ 6, & \text{if } x\in(2,10]\,,\end{cases}$$

$$A_ix=\begin{cases}2+\frac{2}{i}, & \text{if } x\in(0,2]\,,\\ 0, & \text{if } x\in\{0\}\cup(2,10]\,,\end{cases}\qquad \text{for all } i>4.$$

The pair (A_1,S) is weakly compatible, but it is not compatible of type (A), (B), (P) and (C), see [6].

$A_1\,(X)\subset T\,(X)$ and $A_i\,(X)\subset S\,(X)$.

The pair (A_i,T), $i>4$, is weakly compatible because A_i and T commute at their coincidence point $x=0$, but it is not compatible of type (A), (B), (P) and (C).

Let $x_n=2+\frac{1}{n}$. We have $Tx_n=\frac{1}{n}$ and $A_ix_n=0$, hence

$$\lim_{n\to\infty}Tx_n=\lim_{n\to\infty}A_ix_n=t=0.$$

In the other hand, $A_iTx_n=A_i(\frac{1}{n})=2+\frac{2}{i}$ and $TA_ix_n=T0=0$ and so $\lim_{n\to\infty}d\,(A_iTx_n,TA_ix_n)=2+\frac{2}{i}\neq 0$. Therefore, the pair (A_i,T) is not compatible.

$A_i^2x_n=A_i0=0$ and $T^2x_n=T\left(\frac{1}{n}\right)=5+\frac{1}{n}$, so $\lim_{n\to\infty}\left|TA_ix_n-A_i^2x_n\right|=0$ and $\lim_{n\to\infty}\left|A_iTx_n-T^2x_n\right|=\lim_{n\to\infty}(3+\frac{1}{n}-\frac{2}{i})\neq 0$ for all $i>3$. Then, (A_i,T) is not compatible of type (A).

$$\begin{aligned}\lim_{n\to\infty}\left|A_iTx_n-T^2x_n\right|&=3-\frac{2}{i}>\frac{1}{2}\left[\lim_{n\to\infty}\left|A_iTx_n-A_i0\right|+\lim_{n\to\infty}\left|A_i0\;-A_i^2x_n\right|\right]\\&=\frac{1}{2}\left|2+\frac{2}{i}\right|=\frac{1}{i}+1,\end{aligned}$$

hence (A_i,T) is not compatible of type (B).

$\lim_{n\to\infty}\left|A_i^2x_n-T^2x_n\right|=\lim_{n\to\infty}(5+\frac{1}{n})=5\neq 0$. Therefore, (A_i,T) is not compatible of type (P).

$$\begin{aligned}&\lim_{n\to\infty}\left|A_iTx_n-T^2x_n\right|=3-\frac{2}{i}\\&>\frac{1}{3}\left[\lim_{n\to\infty}\left|A_iTx_n-A_i0\right|+\lim_{n\to\infty}\left|A_i0-T^2x_n\right|+\lim_{n\to\infty}\left|A_i0\;-A_i^2x_n\right|\right]\\&=\frac{1}{3}\left(7+\frac{2}{i}\right)\end{aligned}$$

for $i>4$. So, the pair (A_i,T) is not compatible of type (C).

It can be verified that the pairs (A_2,T) , (A_3,T) and (A_4,T) are not weakly compatible because $x=6$ is a coincidence point of A_2 and T, but $A_2T6=4\neq$

$TA_26 = 2$, $x = 7$ is a coincidence point of A_3 and T, but $A_3T(7) = 5 \neq TA_3(7) = 3$ and $x = 8$ is a point of coincidence for A_4 and T, but $A_4T(8) = 6 \neq TA_4(8) = 4$.

Now, we begin to verify the rest of conditions of Corollary 3.2. Let $F(t) = \ln(1+t)$ and $\psi(t) = ht$, where $0 \leqslant h < 1$ and $t > 0$. Set

$$R = \ln(1 + |A_1x - A_iy|) - h \max \left\{ \begin{array}{l} \ln(1 + |Sx - Ty|), \ln(1 + |A_1x - Sx|), \\ \ln(1 + |A_iy - Ty|), \\ \frac{1}{2}\left[\ln(1 + |A_1x - Ty|) + \ln(1 + |Sx - A_iy|)\right] \end{array} \right\}$$

We have the following cases. If $x = 0$ and $y = 0$ we get $R \leqslant 0$ for all $0 \leqslant h < 1$. If $x = 0$ and $y \in (0, 2]$, we get

$$R = \ln\left(3 + \frac{2}{i}\right) - h \max \left\{ \begin{array}{l} \ln(y+6), \ln\left(y + 4 - \frac{2}{i}\right), \\ \frac{1}{2}\left[\ln(y+6) + \ln\left(3 + \frac{2}{i}\right)\right] \end{array} \right\} \leqslant 0$$

for $h \geqslant \frac{\ln\left(3+\frac{2}{i}\right)}{3\ln 2}$ and so there exists $0 \leqslant h < 1$. If $x = 0$ and $y \in (2, 10]$, we get

$$R = -h \max\{\ln(y-1), \ln(y-1), \ln(y-1)\} \leqslant 0$$

for all $0 \leqslant h < 1$. If $x \in (0, 2]$ and $y = 0$, we get

$$R = \ln 4 - h \max \left\{ \begin{array}{l} \ln(x+9), \ln(x+6), \\ \frac{1}{2}\left[\ln 4 + \ln(x+9)\right] \end{array} \right\} \leqslant 0$$

for $h \geqslant \frac{\ln 4}{\ln 11}$ and so there exists $0 \leqslant h < 1$. If $x \in (0, 2]$ and $y \in (0, 2]$, we get

$$R = \ln\left(2 - \frac{2}{i}\right) - h \max \left\{ \begin{array}{l} \ln(x - y + 4), \ln(x+6), \ln\left(y + 4 - \frac{2}{i}\right), \\ \frac{1}{2}\left[\ln(y+3) + \ln\left(x + 7 - \frac{2}{i}\right)\right] \end{array} \right\} \leqslant 0$$

for $h \geqslant \frac{\ln\left(3-\frac{2}{i}\right)}{\ln 8}$. Hence, there exists $0 \leqslant h < 1$. If $x \in (0, 2]$ and $y \in (2, 10]$, we get

$$R = \ln 4 - h \max \left\{ \begin{array}{l} \ln(x + 11 - y), \ln(x+6), \ln(y-1), \\ \frac{1}{2}\left[\ln(|5 - y| + 1) + \ln(x+9)\right] \end{array} \right\} \leqslant 0$$

for $h \geqslant \frac{\ln 4}{\ln 11}$. Hence, there exists $0 \leqslant h < 1$. If $x \in (2, 10]$ and $y = 0$, we get

$$R = -h \max \left\{ \ln(x-1), \ln(x-1), 0, \frac{1}{2}\ln(x-1) \right\} \leqslant 0$$

for all $h \geqslant 0$. Hence, there exists $0 \leqslant h < 1$. If $x \in (2, 10]$ and $y \in (0, 2]$, we get

$$R = \ln\left(3 + \frac{2}{i}\right) - h \max \left\{ \begin{array}{l} \ln(|x - (y+7)| + 1), \ln(x-1), \\ \ln\left(y + 3 - \frac{2}{i}\right), \frac{1}{2}\left[\ln(y+5) + \ln\left(\left|x - 4 - \frac{2}{i}\right| + 1\right)\right] \end{array} \right\} \leqslant 0$$

for $h \geqslant \frac{\ln\left(3+\frac{2}{i}\right)}{\ln 9}$. Hence, there exists $0 \leqslant h < 1$. If x, $y \in (2, 10]$ we get

$$R = -h \max \left\{ \begin{array}{l} \ln(|x - y| + 1), \ln(x-1), \ln(y-1), \\ \frac{1}{2}\left[\ln(y-1) + \ln(x-1)\right] \end{array} \right\} \leqslant 0$$

for all $0 \leqslant h < 1$.

Now, we verify that (A_2, T) and (A_3, T) satisfy all the conditions of Theorem 4.2. Set

$$R_1 = \int_0^{|A_1x-A_2y|} \frac{1}{1+t}\,\mathrm{d}t- \\ - h\max\left\{\begin{array}{l} \int_0^{|Sx-Ty|} \frac{1}{1+t}\,\mathrm{d}t, \int_0^{|A_1x-Sx|} \frac{1}{1+t}\,\mathrm{d}t, \int_0^{|A_2y-Ty|} \frac{1}{1+t}\,\mathrm{d}t, \\ \frac{1}{2}\left[\int_0^{|A_1x-Ty|} \frac{1}{1+t}\,\mathrm{d}t + \int_0^{|Sx-A_2y|} \frac{1}{1+t}\,\mathrm{d}t\right] \end{array}\right\}$$

We have the following cases. If $x = 0$ and $y = 0$ we get $R_1 \leqslant 0$ for all $0 \leqslant h < 1$. If $x = 0$ and $y \in (0, 2]$, we get

$$R_1 = -h\max\left\{\ln(y+6), 0, \ln(y+6), \frac{1}{2}[y+6]\right\} \leqslant 0$$

for all $0 \leqslant h < 1$. If $x = 0$ and $y \in (2, 10]$, we get

$$R_1 = \ln 5 - h\max\left\{\begin{array}{l} \ln(y-1), \ln(|y-6|+1), \\ \frac{1}{2}[\ln(y-1)+\ln 5] \end{array}\right\} \leqslant 0$$

for $h \geqslant \frac{\ln 5}{\ln 9}$, hence there exists $0 \leqslant h < 1$. If $x \in (0, 2]$ and $y = 0$, we get

$$R_1 = \ln 4 - h\max\left\{\begin{array}{l} \ln(x+9), \ln(x+6), 0, \\ \frac{1}{2}[\ln 4+\ln(x+9)] \end{array}\right\} \leqslant 0$$

for all $h \geqslant \frac{\ln 4}{\ln 11}$. Hence, there exists $0 \leqslant h < 1$. If $x \in (0, 2]$ and $y \in (0, 2]$, we get

$$R_1 = \ln 4 - h\max\left\{\begin{array}{l} \ln(4+x-y), \ln(x+6), \ln(y+6), \\ \frac{1}{2}[\ln(y+3)+\ln(x+9)] \end{array}\right\} \leqslant 0$$

for $h \geqslant \frac{\ln 4}{\ln 8}$. Hence there exists $0 \leqslant h < 1$. If $x \in (0, 2]$ and $y \in (2, 10]$, we get

$$R_1 = \ln 2 - h\max\left\{\begin{array}{l} \ln 11, \ln(x+6), \ln(|y-6|+1), \\ \frac{1}{2}[\ln(|5-y|+1)+\ln(x+5)] \end{array}\right\} \leqslant 0$$

for $h \geqslant \frac{\ln 2}{\ln 11}$. Hence,there exists $0 \leqslant h < 1$. If $x \in (2, 10]$ and $y = 0$, we get

$$R_1 = -h\max\left\{\ln(x-1), \ln(x-1), \frac{1}{2}\ln(x-1)\right\} \leqslant 0$$

for all $0 \leqslant h < 1$. In the same manner, if $x \in (2, 10]$ and $y \in (0, 2]$, we get $R_1 \leqslant 0$ for all $0 \leqslant h < 1$. If $x \in (2, 10]$ and $y \in (2, 10]$, we get

$$R_1 = \ln 5 - h\max\left\{\begin{array}{l} \ln(|x-y|+1), \ln(x-1), \ln(|y-6|+1), \\ \frac{1}{2}[\ln(y-1)+\ln|x-6|+1] \end{array}\right\} \leqslant 0$$

for $h \geqslant \frac{\ln 5}{\ln 9}$. Hence, there exists $0 \leqslant h < 1$.Similarly, we can prove the conditions of Theorem 4.2 if we take the mapping A_3 instead of A_2. Finally we remark that all conditions of our theorem are verified and 0 is the unique common fixed point of A_i, S and T.

The following example support our Theorem 3.1.

Remark 3.4. In this example, Theorem 2.6 of [3] is not applicable since the pair (A_2, T) is not weakly compatible, but Theorem 3.1 is applicable. Also, a theorem of [15] for $A_i = A_2$ for all $i \geqslant 2$ is not applicable since the pairs (A_1, S) and (A_2, T) are not compatible. In the same manner, Theorem 1 of [12] is not applicable.

Remark 3.5. In the proof of Lemma 1 of [20] and Theorem 2.1 of [7], the authors applied the inequality

$$a \leqslant b + c \Longrightarrow \int_0^a \varphi(t)dt \leqslant \int_0^b \varphi(t)dt + \int_0^c \varphi(t)dt$$

which is false in general as it is shown by the following example.

Example 3.6. Let $\varphi(t) = t$, $a = 1$, $b = \frac{1}{2}$ and $c = \frac{3}{4}$. Then $1 < \frac{1}{2} + \frac{3}{4}$, but

$$\begin{aligned} \int_0^1 \varphi(t)dt = \frac{1}{2} > \int_0^{\frac{1}{2}} \varphi(t)dt + \int_0^{\frac{3}{4}} \varphi(t)dt \\ = \frac{1}{8} + \frac{9}{32} = \frac{13}{32}. \end{aligned}$$

To correct these errors, the authors should follow the proof of Theorem 2 of [19].

Remark 3.7. In the proof of Theorem 1 of [12], the authors applied the inequality

$$\lim_{n\to\infty} d(x_n, x_{n+1}) = 0 \Longrightarrow \{x_n\} \text{ is a Cauchy sequence}$$

which is false in general. It suffices to take $x_n = \frac{1}{n}$, $n \in \mathbb{N}^*$. Thus, To correct this error, the authors should follow the proof of Theorem 2 of [19].

References

[1] Aliouche, A., A common fixed point theorem for weakly compatible mappings in symmetric spaces satisfying a contractive condition of integral type, *J. Math. Anal. Appl.*, 322 (2) (2006), 796–802.

[2] Aliouche, A., Common fixed point theorems of Gregus type for weakly compatible mappings satisfying generalized contractive conditions, *J. Math. Anal. Appl.*, 341 (1) (2008), 707–719.

[3] Altun, I., Turkoglu, D., Rhoades, B.E., Fixed points of weakly compatible mappings satisfying a general contractive condition of integral type, *Fixed Point Theory And Applications*, Volume 2007 (2007), Article ID 17301, 9 pages.

[4] Branciari, A., A fixed point theorem for mappings satisfying a general contractive condition of integral type, *Int. J. Math. Math. Sci.*, 29 (2002), 531–536.

[5] Djoudi, A., Aliouche, A., Common fixed point theorems of Gregus type for weakly compatible mappings satisfying contractive conditions of integral type, *J. Math. Anal. Appl.*, 329 (1) (2007), 31–45.

[6] Djoudi, A., Merghadi, F., Common fixed point theorems for maps under a contractive condition of integral type, *J. Math. Anal. Appl.*, 341 (2) (2008), 953–960.

[7] Gairola, U.C., Rawat, A.S., A fixed point theorem for integral type inequality, *Int. Journal of Math. Analysis*, Vol. 2, 2008, no. 15, 709–712.

[8] Bouhadjera, H., Djoudi, A., Common fixed point theorems for pairs of single and multivalued D-maps satisfying an integral type, *Annales Math. et Inf.*, 35, (2008), 43–59.

[9] Jungck, G., Compatible mappings and common fixed points, *Int. J. Math. Math. Sci.*, 9 (1986), 771–779.

[10] Jungck, G., Murthy, P.P., Cho, Y.J., Compatible mappings of type (A) and common fixed points, *Math. Japonica.*, 38 (2) (1993), 381–390.

[11] Jungck, G., Common fixed points for non-continuous non-self maps on non metric spaces, *Far East J. Math. Sci.*, 4 (2) (1996), 199–215.

[12] Kohli, J.K., Vashistha, S., Common fixed point theorems for compatible and weakly compatible mappings satisfying general contractive type conditions, *Studii şi Cercetări Ştiinţifice, Seria Matematică*, Universitatea din Bacău, 16 (2006), 33–42

[13] Pant, R.P., Common fixed points of noncommuting mappings, *J. Math. Anal. Appl.*, 188 (1994), 436–440.

[14] Pant, R.P., Common fixed points for four mappings, *Bull. Calcutta. Math. Soc.*, 9 (1998), 281–286.

[15] Pant, R.P., A Common fixed point theorem under a new condition, *Indian J. Pure. Appl. Math.*, 30 (2) (1999), 147–152.

[16] Pathak, H.K., Khan, M.S., Compatible mappings of type (B) and common fixed point theorems of Gregus type, *Czechoslovak Math. J.*, 45 (120) (1995), 685-698.

[17] Pathak, H.K., Cho, Y.J., Kang, S.M., Lee, B.S., Fixed point theorems for compatible mappings of type (P) and applications to dynamic programming, *Le Matematiche*, 1 (1995), 15–33.

[18] Pathak, H.K., Cho, Y.J., Khan, S.M., Madharia, B., Compatible mappings of type (C) and common fixed point theorems of Gregus type, *Demonstratio Math.*, 31 (3) (1998), 499–518.

[19] Rhoades, B.E., Two fixed point theorems for mappings satisfying a general contractive condition of integral type, *Int. J. Math. Math. Sci.*, 63 (2003), 4007–4013.

[20] Vijayaraju, P., Rhoades, B.E., Mohanraj, R., A fixed point theorem for a pair of maps satisfying a general contractive condition of integral type, *Int. J. Math. Math. Sci.*, 15 (2005), 2359–2364.

[21] Sessa, S., On a weak commutativity condition of mappings in fixed point considerations, *Publ. Inst. Math. Beograd.*, 32 (46) (1982), 149–153.

[22] Singh, S.P., Meade, B.A., On common fixed point theorems, *Bull. Austral. Math. Soc.*, 16 (1977), 49–53.

[23] Suzuki, T., Meir-Keeler contractions of integral type are still Meir-Keeler contractions, *Int. J. Math. Math. Sci.*, 2007, Article ID 39281, 6 pages, 2007. doi:10.1155/2007/39281.

[24] Zhang, X., common fixed point theorems for some new generalized contractive type mappings, *J. Math. Anal. Appl.*, 333 (2) (2007), 780–786.

A. Aliouche
Department of Mathematics
University of Larbi Ben M'Hidi Oum-El-Bouaghi
04000
Algeria
e-mail: alioumath@yahoo.fr

F. Merghadi
Department of Mathematics
University of Tebessa
12000
Algeria
e-mail: faycel_mr@yahoo.fr

Annales Mathematicae et Informaticae
36 *(2009) pp. 15–28*
http://ami.ektf.hu

Approximation approach to performance evaluation of Proxy Cache Server systems

Tamás Bérczes

Department of Informatics Systems and Networks, University of Debrecen

Submitted 8 January 2009; Accepted 15 April 2009

Abstract

In this paper we treat a modification of the performance model of Proxy Cache Servers to a more powerful case when the inter-arrival times and the service times are generally distributed. First we describe the original Proxy Cache Server model where the arrival process is a Poisson process and the service times are supposed to be exponentially distributed random variables. Then we calculate the basic performance parameters of the modified performance model using the well known Queueing Network Analysis (QNA) approximation method. The accuracy of the new model is validated by means of a simulation study over an extended range of test cases.

Keywords: Queueing Network, Proxy Cache Server, Performance Models, GI/G/1 queue

1. Introduction

The Internet quickly became an essential and integral part of today's life. However, the booming use of the Web has caused congested networks and overloaded servers. So, the answer from the remote Web server to the client often takes a long time. Adding more network bandwidth is a very expensive solution. From the user's point of view it does not matter whether the requested files are on the firm's computer or on the other side of the world. The main problem is that the same object can be requested by other users at the same time. Because of this situation, identical copies of many files pass through the same network links, resulting in an increased response time. By preventing future transfer, we can cache information and documents that reduces the network bandwidth demand on the external network. In general, there are three types of caches that can be used in isolation or in a hierarchical fashion. Caching can be implemented at browser software [2]; the originating Web sites [3]; and the boundary between the local area network and the

Internet [4]. Browser cache are inefficient since they cache for only one user. Web server caches can improve performance, although the requested files must delivery through the Internet, increasing the response time. In this paper we investigate the third type. Requested documents can be delivered directly from the Web server or through a Proxy Cache Server (PCS). A PCS has the same functionality as a Web server when looked at from the client and the same functionality as a client when looked at from a Web server. The primary function of a PCS is to store documents close to the users to avoid retrieving the same document several times over the same connection. It has been suggested that, given the current state of technology, the greatest improvement in response time will come from installing a PCS at the boundary between the corporate LAN and the Internet.

In this paper, we present an extended version of the performance model of a PCS (see [5, 7]) using a more powerful case when inter-arrival times and the service times are generally distributed.

The organization of the paper is as follows. In Section 2, renewal-based parametric decomposition models are reviewed (see [1, 10]). In Section 3 we introduce a modified version of the original performance model of Proxy Cache Server, where we include the repetition loop at the Proxy Server. A detailed description of the generalized model is given in Section 4. Section 5 is devoted to the validation of the numerical results of the approximation. The paper ends with Comments.

2. The GI/G/1 approximation

The GI/G/1 approximation described here is an example of a method using Parametric Decomposition (see [10]) where the individual queueing nodes are analyzed in isolation based on their respective input and output processes. In this model, the arrival process is a general (GI) arrival process characterised by a mean arrival rate and a squared coefficient of variation (SQV) of the inter-arrival time and the service time may have any general distribution. To use the approximation, we need only to know the mean and the squared coefficient of variance of the inter-arrival times and the service times. In order to apply this method, we assume that the arrival process to a network node is renewal, so the arrival intervals are independent, identically distributed random variables. Immediate feedback, where a fraction of the output of a particular queue enters the queue once again, needs special treatment. Before the detailed analysis of the queueing network is done, the method first removes immediate feedback in a queue by suitably modifying its service time.

This model contains procedures required for modeling of the basic network operations of merging, departure and splitting, arising due to the common sharing of the resources and routing decisions in the network. Futhermore, the approximation provide performance measures (i.e. mean queue lengths, mean waiting times, etc.) for both per-queue and per-network.

The parameters required for the approximation: Arrival process: (λ_A - the mean arrival rate), (c_A^2 - the SQV of the inter-arrival time) and service time (τ_S

- the mean service time), and (c_S^2 - the SQV of the service time) at a considered node.

The approximation method that transforms the two parameters of the internal flows for each of the three basic network operations and the removal of the immediate feedback, as given in [1], is described in the following:

1) *Merging GI traffic flows:* The superposed process of n individual GI flows, each characterized by λ_j and c_j^2 (j $= 1, \ldots, n$), as it enters the considered node is approximated by a GI traffic flow with parameters λ_A and c_S^2, representing the mean arrival rate and SQV of the inter-arrival time of the superposed flow, respectively. The mean arrival rate and the SQV of the inter-arrival time of the superposed flow is given by:

$$\lambda_A = \sum_{j=1}^{n} \lambda_j,$$

$$c_A^2 = \varpi \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_A} c_j^2 + 1 - \varpi,$$

with

$$\varpi = \frac{1}{1 + 4(1-\rho)^2(\nu - 1)},$$
$$\nu = \frac{1}{\sum_{j=1}^{n} \left(\frac{\lambda_j}{\lambda_A}\right)^2},$$

and ρ is the utilisation at the node, defined by $\rho = \lambda_A \tau_S$.

2) *Departure flow from a queue:* The departure flow from a queue is approximated as a GI traffic flow, characterized by λ_D and c_D^2, representing the mean departure rate and SQV of the inter-departure time of the departure flow, respectively. Under equilibrium conditions, the mean flow entering a queue is always equal to the mean flow existing the queue: $\lambda_D = \lambda_A$. The SQV of inter-departure time of the departure flow is given by:

$$c_D^2 = \rho^2 c_S^2 + \left(1 - \rho^2\right) c_A^2$$

3) *Splitting a GI flow Probabilistically:* If a GI flow with parameters λ and c^2 is split into n flows, each selected independently with probability p_i, the parameters for the i-th flow will be given by:

$$\lambda_i = p_i \lambda,$$
$$c_i^2 = p_i c^2 + (1 - p_i).$$

4) *Removing immediate feedback:* If the output traffic from a queue is fed back to this queue itself (Q_i), so that the net arrival process is the sum of the external

arrivals Λ and the fed back portion $p_{ii}\lambda_i$. The approach followed to eliminate this immediate feedback at the queue is to suitably adjust the service time at the queue and the SQV of service time. Assume that the original service parameters at the considered node are: $\tau_{S,U}$ - the mean service time, and $c^2_{S,U}$ - the SQV of the service time. Removing the immediate feedback from that node we will get the modified service parameters:

$$\begin{aligned} \tau_{S,M} &= \frac{\tau_{S,U}}{1-p_{ii}}, \\ c^2_{S,M} &= p_{ii} + (1-p_{ii})c^2_{S,U}, \\ W_{q,M} &= \frac{W_{q,M}}{1-p_{ii}}. \end{aligned}$$

This reconfigured queue without immediate feedback is used subsequently for solving the queueing network.

5) *Mean waiting time:* If the considered node is a GI/G/1 queue, the following Kramer and Langenbach-Belz approximation is used (see [12]):

$$W_q = \frac{\tau_S \cdot \rho(c^2_A + c^2_D)\beta}{2(1-\rho)}$$

with

$$\beta_{Web} = \begin{cases} \exp\left(\frac{2(1-\rho)(1-c^2_A)^2}{3\rho(c^2_A+c^2_D)}\right), & \text{for } c^2_A < 1, \\ 1 & \text{for } c^2_A \geqslant 1. \end{cases}$$

3. The model of Proxy Cache Server

In this section we modified the original (M/M/1) performance model of Proxy Cache Server (see [5]). In this version of the performance model the Proxy Cache Server behaves like a Web server. So, if the size of the file that will pass through the server, is greater then the server's output buffer it will start a looping process until the delivery of all file's is completed (see [11, 6]).

Using Proxy Cache Server, if any information or file is requested to be downloaded, first it is checked whether the document exists on the Proxy Cache Server or not. (We denote the probability of this existence by p). If the document can be found on the PCS then its copy is immediately transferred to the user. In the opposite case the request will be sent to the remote Web server. After the requested document arrived back to the PCS then a copy of it is delivered to the user.

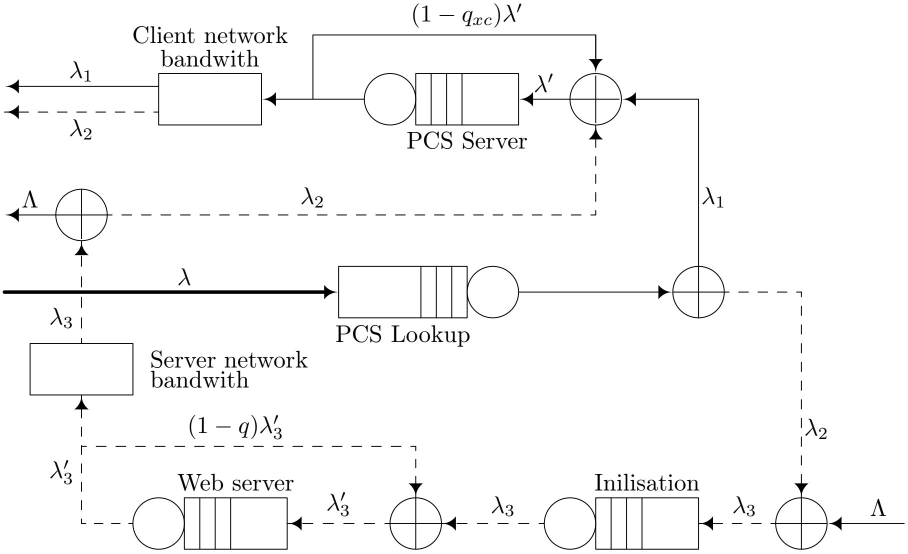


Figure 1: Network model

Figure 1 illustrates the path of a request in the original model (with feedback) starting from the user and finishing with the return of the answer to the user. The notations of the most important basic parameters used in this model are collected in Table 3.

In this section we assume that the requests of the PCS users arrive according to a Poisson process with rate λ, and the external requests arrive to the Web server according to a Poisson process with rate Λ, respectively.

The service rate of the Web server is given by:

$$\mu_{Web} = \frac{1}{Y_S + \frac{B_S}{R_S}}$$

where B_s is the capacity of the output buffer, Y_s is the static server time, and R_s is the dynamic server rate.

The service rate of the PCS is given by the equation:

$$\mu_{PCS} = \frac{1}{Y_{xc} + \frac{B_{xc}}{R_{xc}}}$$

where B_{xc} is the capacity of the output buffer, Y_{xc} is the static server time of the PCS, and R_s is the dynamic server rate of the PCS. The solid line in Figure 1 ($\lambda_1 = p\lambda$) represents the traffic when the requested file is available on the PCS and can be delivered directly to the user. The $\lambda_2 = (1 - p)\lambda$ traffic depicted by dotted line, represents those requests which could not be served by the PCS, therefore these requests must be delivered from the remote Web server. $\lambda_3 = \lambda_2 + \Lambda$ is the flow of the overall requests arriving to the remote Web server. First the λ_3

traffic undergoes the process of initial handshaking to establish a one-time TCP connection (see [11, 7]). We denote by I_s this initial setup.

If the size of the requested file is greater then the Web server's output buffer it will start a looping process until the delivery of all requested file's is completed. Let

$$q = \min\left(1, \frac{B_s}{F}\right)$$

be the probability that the desired file can be delivered at the first attempt. So λ_3' is the flow of the requests arriving at the Web service considering the looping process. According to the conditions of equilibrium and the flow balance theory of queueing networks

$$\lambda_3 = q\lambda_3'$$

Also, the PCS have to be modeled by a queue whose output is redirected with probability $1 - q_{xc} = \min\left(1, \frac{B_{xc}}{F}\right)$ to its input, so

$$\lambda = q_{xc}\lambda'$$

where λ' is the flow of the requests arriving to the PCS, considering the looping process.

Then we get the overall response time (see [5]):

$$\begin{aligned} T_{xc} = {} & \frac{1}{\frac{1}{I_{xc}} - (\lambda)} + p\left\{ \frac{\frac{F}{B_{xc}}}{\frac{1}{\left(Y_{xc}+\frac{B_{xc}}{R_{xc}}\right)} - \frac{\lambda}{q_{xc}}} + \frac{F}{N_c} \right\} \\ & + (1-p)\left\{ \frac{1}{\frac{1}{I_s} - \lambda_3} + \frac{\frac{F}{B_s}}{\frac{1}{\left(Y_s+\frac{B_s}{R_s}\right)} - \frac{\lambda_3}{q}} \right. \\ & \left. + \frac{F}{N_s} + \frac{\frac{F}{B_{xc}}}{\frac{1}{\left(Y_{xc}+\frac{B_{xc}}{R_{xc}}\right)} - \frac{\lambda}{q_{xc}}} + \frac{F}{N_c} \right\}. \end{aligned} \tag{3.1}$$

4. The GI/G/1 model of Proxy Cache Server

In this section instead of M/M/1 queues we will use GI/G/1 queues using the approximation describe in Section 2.

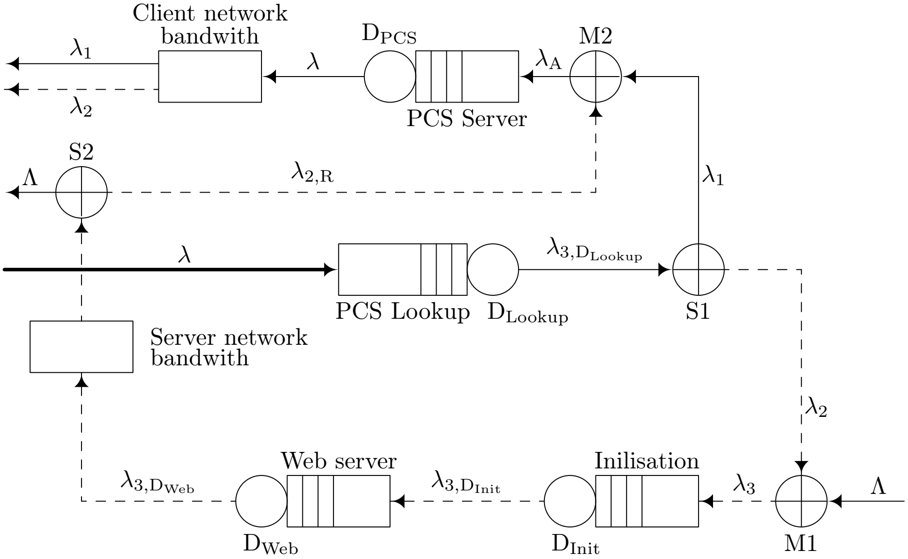


Figure 2: Modified Network model

The requests of the PCS users is assumed to be generalized inter-arrival (GI) process (see [10]) with λ mean arrival rate and with c_λ^2 SQV of the inter-arrival time, and the external arrivals at the remote Web server are generalized inter-arrival process too with parameters Λ and c_Λ^2.

The parameters of the queue PCS Lookup and the queue of the TCP initialization (see Figure 2) are μ_{Lookup}, c^2_{Lookup} and μ_{Init}, c^2_{Init} and the parameters of the Web and PCS servers are μ_{Web}, c^2_{Web} and μ_{PCS}, c^2_{PCS} where:

$$\mu_{Lookup} = \frac{1}{I_{xc}},$$

and

$$\mu_{Init} = \frac{1}{I_S},$$

$$\mu_{Web} = \frac{1}{Y_S + \frac{B_S}{R_S}},$$

$$\mu_{PCS} = \frac{1}{Y_{xc} + \frac{B_{xc}}{R_{xc}}}$$

where I_{xc} is the lookup time of the PCS (in second) and I_s is the TCP setup time. The B_s and B_{xc} parameters are the capacity of the output buffer of the Web server and the PCS, Y_s and Y_{xc} are the static server times, and R_s and R_{xc} are the dynamic server rates for the Web and Proxy servers (see [7]).

In the first step we removed the immediate feedbacks from the Web server and from the PCS, respectively. Figure 2 shows the modified model. After the removal we have to modify the parameters of the corresponding servers:

$$\begin{aligned}\mu_{Web,M} &= \mu_{Web} q,\\ c^2_{Web,M} &= (1-q) + q c^2_{Web},\\ \mu_{PCS,M} &= \mu_{PCS} q_{xc},\\ c^2_{PCS,M} &= (1-q_{xc}) + q_{xc} c^2_{PCS}.\end{aligned}$$

In the model we have 2 superposition point (S1,S2), 2 merging point (M1,M2) and 4 separate queue where we have to recalculate the basic parameters. In S1 position the flow of the requests split in two flows with probability p and $1-p$.

The recalculated parameters of the departure flow after checked of the availability of the required file are:

$$\begin{aligned}\lambda_D &= \lambda,\\ c^2_{D_{Lookup}} &= \rho^2 c^2_{Lookup} + \left(1-\rho^2\right) c^2_\lambda,\end{aligned}$$

where

$$\rho = \frac{\lambda}{\mu_{I_{xc}}}.$$

The solid line (λ_1) represents those requests, which are available on the PCS and can be delivered directly to the user. The λ_2 traffic depicted by dotted line, represents those requests which could not be served by the PCS, therefore these requests must be delivered from the remote Web server.

The parameters of the two flows are:

$$\begin{aligned}\lambda_1 &= p\lambda_D,\\ c^2_1 &= p c^2_{D_{Lookup}} + (1-p),\\ \lambda_2 &= (1-p)\lambda_D,\\ c^2_2 &= (1-p) c^2_{D_{Lookup}} + p.\end{aligned}$$

In M1 position the λ_2 flow and the external requests are merging, and we get the λ_3 flow with the parameters defined below:

$$\begin{aligned}\lambda_3 &= \lambda_2 + \Lambda,\\ c^2_3 &= w\left(\frac{\lambda_2}{\lambda_3} c^2_2 + \frac{\Lambda}{\lambda_3} c^2_\Lambda\right) + (1-w)\end{aligned}$$

where

$$\begin{aligned}w &= \frac{1}{4(1-\rho)^2(\nu-1)},\\ \nu &= \frac{1}{\left(\frac{\lambda_2}{\lambda_3}\right)^2 + \left(\frac{\Lambda}{\lambda_3}\right)^2},\end{aligned}$$

and

$$\rho = \frac{\lambda_3}{\mu_{Init}}$$

The parameters of the departure flow ($\lambda_{3,D_{Init}}$) after the TCP initialisation are:

$$\lambda_{3,D_{Init}} = \lambda_3,$$

$$c^2_{3,D_{Init}} = \rho^2 c^2_{Init} + \left(1 - \rho^2\right) c^2_3,$$

where

$$\rho = \frac{\lambda_{3,D_{Init}}}{\mu_{Init}}.$$

The parameters of the departure flow ($\lambda_{3,D_{Web}}$) from the Web server are:

$$\lambda_{3,D_{Web}} = \lambda_3,$$

$$c^2_{3,D_{Web}} = \rho^2 c^2_{Web,M} + \left(1 - \rho^2\right) c^2_{3,D_{Init}},$$

where

$$\rho = \frac{\lambda_3}{\mu_{Web,M}}.$$

Then in S2 position the $\lambda_{3,D_{Web}}$ flow splits into two parts. One part is the traffic of the external requests, with probability $\frac{\Lambda}{\lambda_2+\Lambda}$, and the second part is the flow ($\lambda_{2,R}$) of the returning requests to the PCS. The parameters of the $\lambda_{2,R}$ traffic are:

$$\lambda_{2,R} = \lambda_2,$$

$$c^2_{2,R} = \frac{\lambda_2}{\lambda_2 + \Lambda} c^2_{3,D_{Web}} + \left(1 - \frac{\lambda_2}{\lambda_2 + \Lambda}\right)$$

In M2 position the λ_1 traffic and the $\lambda_{2,R}$ traffic are merging into λ_A traffic which is described by parameters:

$$\lambda_A = \lambda_1 + \lambda_{2,R} = \lambda_1 + \lambda_2 = \lambda,$$

$$c^2_A = w\left(\frac{\lambda_1}{\lambda} c^2_1 + \frac{\lambda_2}{\lambda} c^2_{2,R}\right) + (1 - w)$$

where

$$w = \frac{1}{4(1-\rho)^2(\nu - 1)},$$

$$\nu = \frac{1}{\left(\frac{\lambda_1}{\lambda}\right)^2 + \left(\frac{\lambda_2}{\lambda}\right)^2},$$

and

$$\rho = \frac{\lambda}{\mu_{PCS,M}}$$

The overall response time can be calculated as follows (see [5, 11]):

$$\begin{aligned}T_{xc} = T_{Lookup} &+ p\left\{T_{PCS} + \frac{F}{N_c}\right\}\\ &+ (1-p)\left\{T_{Init} + T_{Web} + \frac{F}{N_s} + T_{PCS} + \frac{F}{N_c}\right\},\end{aligned}$$

where

$$\begin{aligned}T_{Lookup} &= W_{Lookup} + \frac{1}{\mu_{Lookup}} =\\ &= \frac{\frac{1}{\mu_{Lookup}}\rho_{Lookup}\left(c_\lambda^2 + c_{Lookup}^2\right)\beta}{2\left(1-\rho_{lookup}\right)} + \frac{1}{\mu_{Lookup}},\end{aligned}$$

$$\beta = \begin{cases}\exp\left(-\frac{2\left(1-\rho_{Lookup}\right)\left(1-c_\lambda^2\right)^2}{3\rho_{Lookup}\left(c_\lambda^2+c_{Lookup}^2\right)}\right) & \text{for } c_\lambda^2 < 1\\ 1 & \text{for } c_\lambda^2 \geqslant 1\end{cases}$$

$$\rho_{Lookup} = \frac{\lambda}{\mu_{Lookup}}$$

and

$$\begin{aligned}T_{PCS} &= W_{PCS} + \frac{1}{\mu_{PCS,M}} =\\ &= \frac{\frac{1}{\mu_{PCS,M}}\rho_{pcs}\left(c_A^2 + c_{pcs,M}^2\right)\beta}{2\left(1-\rho_{pcs}\right)} + \frac{1}{\mu_{pcs,M}},\end{aligned}$$

where

$$\beta = \begin{cases}\exp\left(-\frac{2\left(1-\rho_{pcs}\right)\left(1-c_A^2\right)^2}{3\rho_{pcs}\left(c_A^2+c_{pcs,M}^2\right)}\right) & \text{for } c_A^2 < 1\\ 1 & \text{for } c_A^2 \geqslant 1\end{cases}$$

$$\rho_{pcs} = \frac{\lambda_A}{\mu_{pcs,M}}$$

and

$$\begin{aligned}T_{Init} &= W_{Init} + \frac{1}{\mu_{Init}} =\\ &= \frac{\frac{1}{\mu_{Init}}\rho_{Init}\left(c_3^2 + c_{Init}^2\right)\beta_{Init}}{2\left(1-\rho_{Init}\right)} + \frac{1}{\mu_{Init}},\end{aligned}$$

where

$$\beta_{Init} = \begin{cases}\exp\left(-\frac{2\left(1-\rho_{Init}\right)\left(1-c_3^2\right)^2}{3\rho_{Init}\left(c_3^2+c_{Init}^2\right)}\right), & \text{for } c_3^2 < 1\\ 1 & \text{for } c_3^2 \geqslant 1\end{cases}$$

$$\rho_{Init} = \frac{\lambda_3}{\mu_{Init}}$$

and

$$T_{Web} = W_{Web} + \frac{1}{\mu_{Web,M}} =$$

$$= \frac{\frac{1}{\mu_{Web,M}} \rho_{web} \left(c^2_{D_{Init}} + c^2_{web,M} \right) \beta_{web}}{2\left(1 - \rho_{web}\right)} + \frac{1}{\mu_{web,M}},$$

$$\beta_{Web} = \begin{cases} \exp\left(-\frac{2(1-\rho_{web})\left(1 - c^2_{D_{Init}}\right)^2}{3\rho_{web}\left(c^2_{D_{Init}} + c^2_{web,M}\right)} \right), \text{ for } c^2_{D_{Init}} < 1 \\ 1 \qquad\qquad \text{for } c^2_{D_{Init}} \geqslant 1 \end{cases}$$

$$\rho_{web} = \frac{\lambda_3}{\mu_{web,M}}$$

5. Numerical results

For the numerical explorations the corresponding parameters of Cheng and Bose [7] are used. The value of the other parameters for numerical calculations are: $I_s = I_{xc} = 0.004$ seconds, $B_s = B_{xc} = 2000$ bytes, $Y_s = Y_{xc} = 0.000016$ seconds, $R_s = R_{xc} = 1250$ Mbyte/s, $N_s = 1544$ Kbit/s, and $N_c = 128$ Kbit/s. These values are chosen to conform to the performance characteristics of Web servers in [9].

For validating the approximation, we wrote a simulation program in Microsoft Visual Basic 2005 under .NET framework 2.0. It was run on a PC with a T2300 Intel processor (1.66 GHz) with 2 GB RAM. First we validated the simulation program using exponential distributions. For validation we calculated the analytical results of the overall response time given by Eq(3.1) and compared to the simulation results. In Table 1, we can see that the corresponding mean of the total response times are very close to each other; they are the same at least up to the 4th decimal digit.

For the validation of the approximation method we used the following distributions (see [8]). In case $0 < c_X < 1$ we use an $E_{k-1,k}$ distribution, where $\frac{1}{k} \leqslant k < \frac{1}{k-1}$. In this case the approximating $E_{k-1,k}$ distribution is with probability p (resp. $1-p$) the sum of $k-1$ (resp. k) independent exponentials with common mean $\frac{1}{\mu}$. Choosing

$$p = \frac{1}{1+c^2_X}\left(kc^2_X - \left(k\left(1+c^2_X\right) - k^2c^2_X\right)^{1/2} \right) \text{ and } \mu = \frac{k-p}{E(X)}$$

the $E_{k-1,k}$ distribution matches $E(X)$ and c_X.

In case $c_X > 1$ we fits a $H2(p_1; p_2; \mu_1; \mu_2)$ hyper-exponential distribution with balanced mean (see [8]):

$$\frac{p_1}{\mu_1} = \frac{p_2}{\mu_2}$$

So the parameters of this H_2 distributions are:

$$p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{c_X^2 - 1}{c_X^2 + 1}}\right), \quad p_2 = 1 - p_1,$$

and

$$\mu_1 = \frac{2p_1}{E(X)}, \quad \mu_2 = \frac{2p_2}{E(X)}.$$

For the easier understanding we used for all queues the same SQV rate. In Table 2, we show the results of the simulations and approximations with various parameters.

Parameters	Analytical result	Simulation result	Approximation	Difference
$\lambda = 20$, $\Lambda = 100$	0.425793	0.425706	0.425793	0.000087
$\lambda = 80$, $\Lambda = 100$	0.430135	0.430136	0.430135	0.000001

Table 1: **Exponential distribution**

Arrival intensity	SQV	Simulation result	Approx. result	Difference
$\lambda = 20$ $\Lambda = 100$	0.1	0.423084	0.423129	0.000045
	0.8	0.425127	0.425189	0.000062
	1.2	0.423042	0.426328	0.003286
	1.8	0.421744	0.427979	0.006235
$\lambda = 80$ $\Lambda = 100$	0.1	0.423591	0.423974	0.000383
	0.8	0.428624	0.428725	0.000101
	1.2	0.425821	0.431507	0.005686
	1.8	0.424049	0.435869	0.011820

Table 2: **Approximation**

6. Comments

In this paper we modified the performance model of Proxy Cache Server to a more powerful case when the arrival processes is a GI process and the service times may have any general distribution. To obtain the overall response time we used the QNA approximation method, which was validated by simulation. As we can see in Table 2, when the $SQV < 1$ the overall response time obtained by approximation is very close to response time obtained by simulation; they are the same at least up to the 3–4th decimal digit. In case when the $SQV > 1$ the response times are the same only to 2–3th decimal digit. We can see, when the $SQV = 1.2$ the difference between the response times are 0.005686, and when the $SQV = 1.8$ the difference between the response times is greater (0.01182). So, using greater SQV the approximation error is greater.

Table 3: **Notations**

λ:	arrival rate from the PCS
Λ:	external arrival rate
F:	average file size (in byte)
p:	cache hit rate probability
B_{xc}:	PCS output buffer (in byte)
I_{xc}:	lookup time of the PCS (in second)
Y_{xc}:	static server time of the PCS (in second)
R_{xc}:	dynamic server time of the PCS (in byte/second)
N_c:	client network bandwidth (in bit/second)
B_s:	Web output buffer (in byte)
I_s:	lookup time of the Web server (in second)
Y_s:	static server time of the Web server (in second)
R_s:	dynamic server time of the Web server (in byte/second)
N_s:	server network bandwidth (in bit/second)

References

[1] Atov I., QNA Inverse Model for Capacity Provisioning in Delay Constrained IP Networks. *Centre for Advanced Internet Architectures. Technical Report 040611A* Swinburne University of Technology Melbourne, Australia, (2004).

[2] Aggarwal, C., Wolf, J.L. and Yu, P.S., Caching on the World Wide Web. *IEEE Transactions on Knowledge and Data Engineering*, 11 (1999) 94–107.

[3] Almeida, V.A.F., de Almeida, J.M. and Murta, C.S. Performance analysis of a WWW server. *Proceedings of the 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems*, San Diego, USA, December 8. 13 (1996).

[4] Arlitt, M.A. and Williamson, C.L. Internet Web servers: workload characterization and performance implications. *IEEErACM Transactions on Networking,* 5 (1997), 631–645.

[5] Berczes, T. and Sztrik, J., Performance Modeling of Proxy Cache Servers. *Journal of Universal Computer Science.*, 12 (2006) 1139–1153.

[6] Berczes, T., Guta, G., Kusper, G., Schreiner, W. and Sztrik, J., Analyzing Web Server Performance Models with the Probabilistic Model Checker PRISM. *Technical report no. 08-17 in RISC Report Series* , University of Linz, Austria. November 2008.

[7] Bose, I. and Cheng, H.K., Performance models of a firms proxy cache server. *Decision Support Systems and Electronic Commerce.*, 29 (2000) 45–57.

[8] Henk C. Tijms *Stochastic Modelling and Analysis. A computational approuch.* John Wiley & Sons, Inc. New York, (1986).

[9] Menasce, D.A. and Almeida, V.A.F., *Capacity Planning for Web Performance: Metric, Models, and Methods.* Prentice Hall., (1998).

[10] Sanjay K. Bose *An introduction to queueing systems.* Kluwer Academic/Plenum Publishers, New York, (2002).

[11] Slothouber L.P., A model of Web server performance. *5th International World Wide Web Conference, Paris, Farnce.*, (1996).

[12] Whitt, W., The Queueing Network Analyzer. *Bell System Technical Journal.*, Vol. 62, No. 9 (1983) 2799–2815.

Tamás Bérczes
Department of Informatics Systems and Networks
University of Debrecen
P.O. Box 12
H-4010 Debrecen
Hungary
e-mail: `berczes.tamas@inf.unideb.hu`

Annales Mathematicae et Informaticae
36 *(2009) pp. 29–41*
http://ami.ektf.hu

Crossed ladders and Euler's quartic

A. Bremner[a], R. Høibakk[b], D. Lukkassen[bc]

[a]School of Mathematics, Arizona State University, USA
[b]Narvik University College, Norway
[c]Norut Narvik, Norway

Submitted 8 January 2008; Accepted 15 April 2009

Abstract

We investigate a particular form of the classical "crossed ladders" problem, finding many parametrized solutions, some polynomial, and some involving Fibonacci and Lucas sequences. We establish a connection between this particular form and a quartic equation studied by Euler, giving corresponding solutions to the latter.

MSC: 11D25, 11G05.

1. Introduction

The so-called "Crossed Ladders Problem" can be formulated as follows. Two ladders of lengths a,b, lean against two vertical walls as shown in Figure 1. The ladders cross each other at a point with distance c above the ground. Determine the distance x between the walls and the heights y, z, above the ground of the points where the ladders touch the walls.

The defining system of equations is

$$x^2 = a^2 - y^2 = b^2 - z^2, \tag{1.1}$$

$$c = \frac{yz}{y+z}. \tag{1.2}$$

There is enormous literature on the crossed ladder problem, as may be seen for example by consulting the extensive bibliography of Singmaster's "Sources in Recreational Mathematics", Section 6L: "Geometric Recreations. Crossed ladders"; see Singmaster [9]. The second and third authors have also recently investigated this problem; see Høibakk et al. [4, 5].

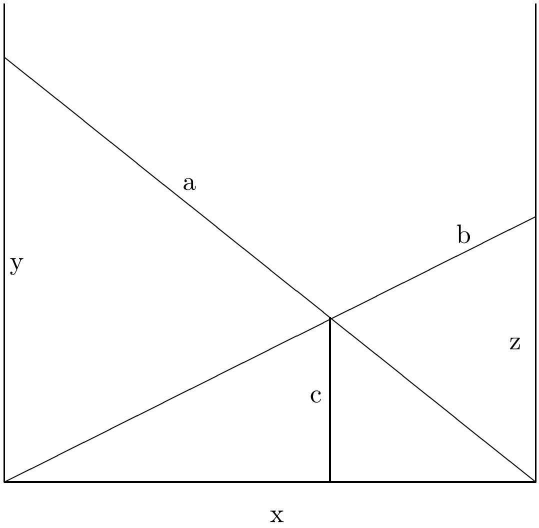


Figure 1: The crossed ladders

Our interest is in ladder problems where the lengths are all integers, and so we are reduced to finding integer solutions to the Diophantine system (1.1) where, without loss of generality, we may assume x, y, z, a, b have no common divisor. For certainly a solution of the ladder problem gives rise to such x, y, z, a, b; and conversely, given coprime x, y, z, a, b, then scaling by the factor

$$S = \frac{y+z}{\gcd(yz, y+z)}$$

results in an integer solution to (1.1), (1.2). Høibakk et al. [4] observe empirically that the ratio $y : z$ frequently takes an integer value greater than 1, and in this note we investigate the conditions and implications that imposing this restriction implies. Putting

$$y = Mz, \qquad M > 1, \tag{1.3}$$

gives

$$x^2 + M^2z^2 = a^2, \qquad x^2 + z^2 = b^2, \tag{1.4}$$

which as the intersection of two quadrics in projective 3-space (with a point at $z = 0$) represents an elliptic curve. A quartic form is easy to derive. At (1.3), (1.4) we can set

$$x : y : z : a : b = (X^2 - Y^2) : 2MXY : 2XY : Z : (X^2 + Y^2), \tag{1.5}$$

with

$$X^4 + (4M^2 - 2)X^2Y^2 + Y^4 = Z^2. \tag{1.6}$$

The inverse transformation is given by

$$\left(\frac{X}{Y},\ \frac{Z}{Y^2}\right) = \left(\frac{x+b}{z},\ \frac{2a(x+b)}{z^2}\right).$$

Thus solutions to the crossed ladder problem under the restriction (1.3) correspond to solutions of the Diophantine equation (1.6). The equation

$$X^4 + mX^2Y^2 + Y^4 = \Box \tag{1.7}$$

has been studied since the 17th century. A solution is said to be trivial if either $XY = 0$ or if $X^2 = Y^2 = 1$, which can occur only when m is of the form $k^2 - 2$. Fermat showed there are no non-trivial solutions for $m = 0$. Euler showed that for $m = 14$ there are only trivial solutions, and found non-trivial solutions for 47 values of m between 2 and 200, and for 73 values of $-m$ between 2 and 200. Pocklington [7], Sinha [10], and Zhang [11] produced classes of m for which (1.7) has no non-trivial solutions; and Brown [3] completed the determination of solvability of (1.7) in the range $0 \leqslant m \leqslant 100$. Bremner & Jones [2] studied the equation in some detail, determining solvability of (1.7) (subject to standard conjectures) in the range $|m| \leqslant 3000$. We can deduce from the tables of [2], for example, that the smallest value of M for which non-trivial solutions of (1.6) exist is $M = 7$, with small solutions at $(X, Y) = (5, 1)$, leading to $(x, y, z, a, b, c) = (12, 35, 5, 74, 26, \frac{35}{8})$, and $(X, Y) = (6, 1)$ with $(x, y, z, a, b, c) = (35, 84, 12, 91, 37, \frac{21}{2})$.

It is our intention here to investigate the quartic (1.6) and derive parametrized families for M for which there exist non-trivial solutions. Surprisingly many values of M for which non-trivial solutions to (1.6) exist turn out to be members of such infinite families. We describe several such families, and indicate the corresponding point on (1.6). It is then straightforward to compute the corresponding solution to the crossed ladder problem by means of the ratios at (1.5). Some of the parametric families are given in terms of polynomials, others in terms of Fibonacci and Lucas sequences.

The ideas are essentially ad hoc, and by no means exhaustive: many values of M for which non-trivial solutions to (1.6) exist have not been found as members of infinite families.

We note that a cubic form of the elliptic curve at (1.4) is given by

$$E : v^2 = u^3 + (-2M^2 + 1)u^2 + (M^4 - M^2)u = u(u - M^2)(u - (M^2 - 1)), \tag{1.8}$$

where the maps are given by

$$x : z : a : b = (v^2 - u^2) : 2uv : (-u^3 + (M^4 - M^2)u) : (v^2 + u^2),$$

and

$$(u, v) = \left(\frac{(a-b)(a-x)}{z^2},\ \frac{(a-b)(b+x)(a-x)}{z^3}\right).$$

We therefore also obtain the maps between the cubic curve at (1.8) and the quartic curve at (1.6), namely

$$(X, Y, Z) = (v,\ u,\ u^3 - (M^4 - M^2)u),$$

$$(u,v)=\left(\frac{X^2+(2M^2-1)Y^2+Z}{2Y^2},\ \frac{X(X^2+(2M^2-1)Y^2+Z)}{2Y^3}\right).$$

2. Linear parametrizations

We show that there can be no nontrivial points on the curve (1.6), where X, Y, Z are linear polynomials in M. The curve (1.8) represents a *rational* elliptic surface S, and as such we know by results of Shioda (see, for example, Shioda [8, Cor. 5.3, Thm. 10.8] that the Mordell-Weil group of (1.8) over $\mathbb{C}(M)$ is generated by those points which are given by polynomials u at most quadratic in M. Finding points on (1.8) over $\mathbb{C}(M)$ whose u-coordinate is at most quadratic in M is a straightforward machine computation. However, it is not necessary to carry out: the discriminant of the cubic model at (1.8) is equal to $16M^4(M^2-1)^2$, so that the curve is singular at $(M), (1/M), (M \pm 1)$. Computing the Kodaira reduction types, the Shioda formula for the rank gives $\text{rank}(E(\mathbb{C}(M))) = 0$, and consequently $\text{rank}(E(\mathbb{Q}(M))) = 0$. The only points on (1.8) are the torsion points, namely $(0,0)$, $(M^2, 0)$, and $(M^2-1, 0)$, corresponding to trivial points on (1.6).

3. Parametrizations of higher degree

At (1.4), we set without loss of generality $b+z = gp^2$, $b-z = gq^2$, $a+y = hr^2$, $a-y = hs^2$, $(x =)gpq = hrs$, $(p,q) = (r,s) = 1$, for integers p, q, r, s and g, h; and the restriction (1.3) demands

$$M=\frac{pq(r^2-s^2)}{rs(p^2-q^2)}\in\mathbb{Z}. \tag{3.1}$$

We correspondingly have crossed ladder solution

$$x:y:z:a:b = 2pq : M(p^2-q^2) : (p^2-q^2) : \frac{pq(r^2+s^2)}{rs} : (p^2+q^2),$$

and point at (1.6) given by

$$(X,Y,Z)=\left(p+q,\ p-q,\ \frac{2pq(r^2+s^2)}{rs}\right).$$

We study several particular cases.

3.1. Case I

We suppose $(r,s) = (pq,1)$, which implies $g = h$ and demands

$$M=\frac{p^2q^2-1}{p^2-q^2}\in\mathbb{Z}. \tag{3.2}$$

An immediate family of solutions arises on setting $(p,q) = (2n+1, 2n-1)$ with corresponding M and point on (1.6) given by:

$$M = n(2n^2-1), \qquad (X,Y,Z) = (2n,\ 1,\ 8n^4-4n^2+1).$$

This gives numerical values of $M = \boxed{14, 51, 124, 245, \ldots}$

The curve E takes the form

$$E_n : v^2 = u\left(u - n^2(2n^2-1)^2\right)\left(u - (n^2-1)(4n^4+1)\right),$$

and the corresponding point on the elliptic curve E_n is

$$P(u,v) = \left((n^2-1)(2n^2-1)^2,\ 2n(n^2-1)(2n^2-1)^2\right).$$

We can compute multiples of P to obtain parametrized solutions to the crossed ladder problem of increasing degree. For example,

$$2P = \left(\frac{(8n^4-4n^2+1)^2}{16n^2},\ -\frac{(16n^4-1)(8n^4-4n^2+1)}{64n^3}\right),$$

corresponding to the crossed ladder solution

$$\begin{aligned}(x,y,z,a,b) = ((4n^2-2n-1)(4n^2+2n-1)(8n^3-2n-1)(8n^3-2n+1),\\ 8n^2(2n^2-1)(16n^4-1)(8n^4-4n^2+1),\\ 8n(16n^4-1)(8n^4-4n^2+1),\\ 2048n^{12}-2048n^{10}+896n^8-384n^6+128n^4-16n^2+1,\\ -(1024n^{10}-768n^8+512n^6-160n^4+16n^2+1)).\end{aligned}$$

(Remark: the torsion group on E_n is $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ and points $Q+T$ for T torsion return the same values of (x,y,z,a,b) as for Q, up to sign; so it is only of interest to consider (x,y,z,a,b) corresponding to the direct multiples of P on E_n). The point $3P$ returns polynomials of degree 24.

We can obtain families of solutions to (3.2) by demanding that

$$n(pq+\epsilon) = p^2 - q^2, \qquad \epsilon = \pm 1, \tag{3.3}$$

for integers n. When $n = 1$, the theory of the Pell equation gives all solutions of (3.3) as $(p,q) = (F_{k+1}, F_k)$, where F_i is the i-th Fibonacci number. In this instance, $M = F_{k+1}F_k - (-1)^k$, with associated ladder solution

$$(x,y,z,a,b) = (2F_{k+1}F_k,\ F_{k+1}^2F_k^2-1,\ F_{k+1}^2-F_k^2,\ F_{k+1}^2F_k^2+1,\ F_{k+1}^2+F_k^2),$$

and point on (1.6)

$$M = F_{k+1}F_k - (-1)^k, \qquad (X,Y,Z) = (F_{k+2},\ F_{k-1},\ 2(F_{k+1}^2F_k^2+1)).$$

Numerical values of M that occur are $\boxed{M = 7, 14, 41, 103, 274, \ldots}$

When $n > 1$, solutions are provided in terms of the recurrence relation

$$R_i = nR_{i-1} + R_{i-2}, \quad i \geqslant 2, \qquad R_0 = \epsilon, R_1 = 1, \quad \epsilon = \pm 1.$$

Then

$$R_{k+1}^2 - R_k^2 = n(R_{k+1}R_k - \epsilon(-1)^k),$$

and taking $(p, q) = (R_{k+1}, R_k)$ gives rise to

$$M = \frac{1}{n}(R_{k+1}R_k + \epsilon(-1)^k).$$

We have

$$R_{2i+1} \equiv 1 \bmod n, \qquad R_{2i} \equiv \epsilon \bmod n,$$

so M will be integral precisely when k is odd. Thus setting $(p, q) = (R_{2i}, R_{2i-1})$ gives rise to

$$M = \frac{1}{n}(R_{2i}R_{2i-1} - \epsilon) \in \mathbb{Z}.$$

The corresponding point on (1.6) is given by

$$(X, Y, Z) = (R_{2i} + R_{2i-1},\ R_{2i} - R_{2i-1},\ 2(R_{2i}^2 R_{2i-1}^2 + 1)).$$

The case $n = 2$, $\epsilon = 1$, gives the well-known Pell sequence $R = \{1, 1, 3, 7, 17, 41, \ldots\}$ with corresponding M equal to $\boxed{59, 2029, \ldots}$. When $\epsilon = -1$, then the R-sequence is $\{-1, 1, 1, 3, 7, 17, 41, \ldots\}$ with corresponding M equal to $\boxed{11, 349, \ldots}$.

If we leave n as parameter, then we obtain the following values of M:

$$n^4 + 2\epsilon n^3 + 4n^2 + 4\epsilon n + 3 \tag{3.4}$$

$$n^8 + 2\epsilon n^7 + 8n^6 + 12\epsilon n^5 + 21n^4 + 22\epsilon n^3 + 20n^2 + 12\epsilon n + 5 \tag{3.5}$$

$$\vdots$$

Without loss of generality (by changing the sign of n if necessary) we may take $\epsilon = 1$, and the first line above corresponds to the crossed ladder problem solution given by

$$\begin{aligned}
&(x, y, z, a, b) = \\
&(2(n^2 + n + 1)(n^3 + n^2 + 2n + 1), \\
&n(n + 1)(n^2 + 1)(n^2 + n + 2)(n^4 + 2n^3 + 4n^2 + 4n + 3), \\
&n(n + 1)(n^2 + 1)(n^2 + n + 2), \\
&n^{10} + 4n^9 + 12n^8 + 24n^7 + 38n^6 + 46n^5 + 44n^4 + 32n^3 + 17n^2 + 6n + 2, \\
&n^6 + 2n^5 + 6n^4 + 8n^3 + 9n^2 + 6n + 2),
\end{aligned}$$

with point on (1.6) given by

$$(X, Y, Z) = ((n + 1)(n^2 + n + 2),\ n(n^2 + 1),$$

$$2(n^{10}+4n^9+12n^8+24n^7+38n^6+46n^5+44n^4+32n^3+17n^2+6n+2)).$$

Other approaches to making the quotient (3.2) integral include setting $p = F_{n+1} + F_{n-1}$, $q = F_n$, when we obtain

$$M = \frac{1}{4}(F_{n+2} + F_n)(F_n + F_{n-2}),$$

which is integral precisely when $n \equiv \pm 2 \bmod 6$. Numerical values of M are given by $M = \boxed{11, 551, \ldots}$. The point on (1.6) is

$$(X, Y, Z) = (F_{n+1}, F_{n-1}, \frac{1}{2}F_{2n+1}F_{2n-1}).$$

We can also take $(p, q) = (P_n, Q_n)$, where P_n, Q_n are the familiar Pell-sequences defined by $P_n = 2P_{n-1} + P_{n-2}$, $n \geqslant 2$, $P_0 = 1$, $P_1 = 1$, and $Q_n = 2Q_{n-1} + Q_{n-2}$, $n \geqslant 2$, $Q_0 = 0$, $Q_1 = 1$. Thus P is the sequence $1, 1, 3, 7, 17, 41, 99, \cdots$, Q the sequence $0, 1, 2, 5, 12, 29, 70, \cdots$, and $P_n^2 - 2Q_n^2 = (-1)^n$. Then

$$M = 2Q_n^2 - (-1)^n = P_{n+1}P_{n-1},$$

with numerical values $M = \boxed{7, 51, 287, \ldots}$. The point on (1.6) is

$$(X, Y, Z) = (P_n + Q_n, P_n - Q_n, 2(P_n^2Q_n^2 + 1)) = (Q_{n+1}, Q_{n-1}, 2(P_n^2Q_n^2 + 1)).$$

3.2. Case II

We assume $p^2 - q^2 = r^2 - s^2$, $q = rs$, which implies $M = p$, and demands

$$p^2 - r^2(s^2 + 1) = -s^2. \tag{3.6}$$

We consider s to be a fixed parameter, and by considering norms from the quadratic field $\mathbb{Q}(\sqrt{s^2+1})$ in which we note $s + \sqrt{s^2+1}$ is a unit of norm -1, can define solutions (p_i, r_i) by means of

$$p_i + r_i\sqrt{s^2+1} = (s + \sqrt{s^2+1})^{2i}(p_0 + r_0\sqrt{s^2+1})$$

for an initial solution (p_0, r_0). It is readily seen that p_i and r_i are determined recursively by

$$\begin{aligned} p_{i+2} &= (4s^2+2)p_{i+1} - p_i, & p_1 &= (2s^2+1)p_0 + 2s(s^2+1)r_0, \\ r_{i+2} &= (4s^2+2)r_{i+1} - r_i, & r_1 &= 2sp_0 + (2s^2+1)r_0. \end{aligned}$$

The crossed ladder solution is

$$(x, y, z, a, b) = (2p_ir_is,\ p_i(p_i^2 - r_i^2s),\ p_i^2 - r_i^2s^2,\ p_i(r_i^2 + s^2),\ p_i^2 + r_i^2s^2),$$

and the corresponding point on (1.6) is given by:

$$M = p_i, \qquad (X, Y, Z) = (p_i + r_is,\ p_i - r_is,\ 2p_i(r_i^2 + s^2)).$$

Taking $(p_0, r_0) = (1, 1)$, then $(p_1, s_1) = (2s^3 + 2s^2 + 2s + 1,\ 2s^2 + 2s + 1)$, and we obtain the sequence p_i, $i \geqslant 1$ as:

$$1 + 2s + 2s^2 + 2s^3,\ 1 + 4s + 8s^2 + 12s^3 + 8s^4 + 8s^5,\ \dots$$

As an example, the former corresponds to crossed ladder solution

$$\begin{aligned}(x, y, z, a, b) =&(2s(1 + 2s + 2s^2)(1 + 2s + 2s^2 + 2s^3),\\ &(1 + s)(1 + 2s)(1 + s + 2s^2)(1 + 2s + 2s^2 + 2s^3),\\ &(1 + s)(1 + 2s)(1 + s + 2s^2),\\ &(1 + 2s + 2s^2 + 2s^3)(1 + 4s + 9s^2 + 8s^3 + 4s^4),\\ &1 + 4s + 9s^2 + 16s^3 + 20s^4 + 16s^5 + 8s^6),\end{aligned}$$

and point on (1.6) with $M = 2s^3 + 2s^2 + 2s + 1$,

$$(X, Y, Z) = ((2s+1)(2s^2+s+1),\ s+1,\ 2(2s^3+2s^2+2s+1)(4s^4+8s^3+9s^2+4s+1)).$$

Numerical values of M that arise from these parametrizations are

$$\boxed{M = 7, 11, 29, 41, 79, 103, 169, 199, 209, \dots}$$

(with $M = 199$ arising from $M = 1 + 4s + 8s^2 + 12s^3 + 8s^4 + 8s^5$, the other values arising from $M = 1 + 2s + 2s^2 + 2s^3$).

If instead we take $(p_0, r_0) = (\pm s^2, s)$, then the resulting sequence p_i is

$$s^2(3 + 4s^2),\ s^2(5 + 20s^2 + 16s^4),\ \dots$$

which is a special case of the sequence derived under Case III, and is not considered further here.

If at (3.6) we consider instead r to be a fixed parameter, then

$$p^2 - s^2(r^2 - 1) = r^2. \tag{3.7}$$

Analogously,

$$p_i + s_i\sqrt{r^2 - 1} = (r + \sqrt{r^2 - 1})^i(p_0 + s_0\sqrt{r^2 - 1}),$$

for an initial solution (p_0, s_0). Taking $(p_0, s_0) = (r, 0)$ we obtain the recurrences

$$\begin{aligned}p_{i+2} &= 2rp_{i+1} - p_i, \qquad p_0 = r, \quad p_1 = r^2,\\ s_{i+2} &= 2rs_{i+1} - s_i, \qquad s_0 = 0, \quad s_1 = r,\end{aligned}$$

giving the sequence of p_i (and hence M) as

$$r(2r^2 - 1), \quad r^2(4r^2 - 3), \quad r(8r^4 - 8r^2 + 1), \quad \dots$$

Numerical values of M arising from these parametrizations are:

$$\boxed{M = 14, 51, 52, 124, 194, 245, \dots}$$

3.3. Case III

We suppose $(r, s) = (p, 1)$, which demands

$$\frac{q(p^2-1)}{p^2-q^2} \in \mathbb{Z}. \tag{3.8}$$

Solutions are generated by the recurrence relations

$$p_i = np_{i-1} - p_{i-2}, \quad i \geqslant 3, \qquad p_1 = -1,\ p_2 = 1,$$

$$q_i = nq_{i-1} - q_{i-2}, \quad i \geqslant 3, \qquad q_1 = 1,\ q_2 = 1,$$

where, on taking $(p, q) = (p_i, q_i)$, we have corresponding value of M equal to $(\frac{n+2}{4})q_i$. Accordingly, we take $n \equiv 2 \bmod 4$. The first three values of (p_i, q_i) with the corresponding M and point (X, Y, Z) on (1.6) are as follows, where we require $n \equiv 2 \bmod 4$.

$$\begin{gathered}(p_3, q_3) = (n+1, n-1), \quad M = \left(\frac{n+2}{4}\right)(n-1),\\ (X, Y, Z) = \left(n, 1, (n-1)\left(\frac{1}{2}n^2 + n + 1\right)\right);\end{gathered} \tag{3.9}$$

$$\begin{gathered}(p_4, q_4) = (n^2-n+1, n^2-n-1), \quad M = \left(\frac{n+2}{4}\right)(n^2-n-1),\\ (X, Y, Z) = \left(n^2-1, n, \frac{1}{2}(n^2-n-1)(n^4+2n^3-n^2-2n+2)\right);\end{gathered}$$

$$\begin{gathered}(p_5, q_5) = (n^3+n^2-2n-1, n^3-n^2-2n+1),\\ M = \left(\frac{n+2}{4}\right)(n^3-n^2-2n+1),\\ (X, Y, Z) =\\ \left(n(n^2-2), n^2-1, \frac{1}{2}(n^3-n^2-2n+1)(n^6+2n^5-3n^4-6n^3+2n^2+4n+2)\right).\end{gathered}$$

Numerical values of M arising from these parametrizations are:

$$\boxed{M = 7, 10, 22, 27, 41, 45, 52, 58, 76, 85, 115, 126, 162, 175, 217, \ldots}$$

The solution at (3.9) has M quadratic in the parameter n, and we can find all the corresponding parametrizations of (1.6) because it may be shown that the curve (1.8) which equals

$$E_n : v^2 = u\left(u - \frac{1}{16}(n-1)^2(n+2)^2\right)\left(u - \frac{1}{16}(n-2)(n+3)(n^2+n+2)\right),$$

is of rank 1 over $\mathbb{Q}(n)$ with generator

$$P(u,v) = \left(\frac{1}{16}(n-2)(n-1)(n^2+n+2),\ \frac{1}{16}n(n-2)(n-1)(n^2+n+2)\right). \tag{3.10}$$

(That the rank is 1 follows from Shioda's formula for the K3 elliptic surface represented by E_n; that P is a generator follows from computing its height of 7/8, and using arguments similar to those of Kuwata [6]). So, for example,

$$2P = \left(\frac{(n-1)^2(n^2+2n+2)^2}{16n^2},\ -\frac{(n-1)^2(n+1)(n^2+1)(n^2+2n+2)}{16n^3}\right),$$

leading to

$$(X,Y,Z) = \Big(-(n+1)(n^2+1),\ n(n^2+2n+2),\\ \frac{1}{2}(n^8+4n^7+6n^6+4n^5-n^4-4n^3+2n^2+4n+2)\Big),$$

and

$$3P = \left(\frac{(n-2)(n-1)(n^2+n+2)(n^4+3n^3+3n^2+n+1)^2}{16(n^4+n^3+n^2-n-1)^2},\right.\\ \left.\frac{(n-2)(n-1)n(n^2+n+2)(n^3-n-1)(n^3+2n^2+3n+3)(n^4+3n^3+3n^2+n+1)}{16(n^4+n^3+n^2-n-1)^3}\right),$$

leading to

$$(X,Y,Z) = \\ = \Big(n(n^3-n-1)(n^3+2n^2+3n+3),\ (n^4+3n^3+3n^2+n+1)(n^4+n^3+n^2-n-1),\\ -\frac{1}{2}(n-1)(n^2+2n+2)(n^{14}+6n^{13}+17n^{12}+36n^{11}+66n^{10}+104n^9+139n^8\\ +140n^7+95n^6+38n^5+4n^4+6n^2+4n+1)\Big).$$

An alternative approach to making (3.8) integral is to set $p = F_n$, $q = F_{n+1}+F_{n-1}$, where n is odd (so that $F_n^2 - 1 = F_{n+1}F_{n-1}$). Then $M = \frac{1}{4}(F_{n+1}+F_{n-1})$, and is integral precisely when $n \equiv 3 \bmod 6$. This gives rise to numerical values for M equal to $M = \boxed{19, 341, \ldots}$. Alternatively, setting $p = F_{n+1}+F_{n-1}$, $q = F_n$, and using that $(F_{n+1}+F_{n-1})^2 - 1 = 5F_{n+1}F_{n-1}$ for n odd, then $M = \frac{5}{4}F_n$, which is integral precisely when $n \equiv 0 \bmod 6$. Corresponding numerical values are $M = \boxed{10, 180, 3230, \ldots}$

3.4. Case IV

We demand $p+q = r-s$, $pq = rs$, by setting

$$p = n+1,\ q = n-1,\ r = m+n,\ s = m-n,$$

where

$$m^2 - 2n^2 = -1, \tag{3.11}$$

in which case $M = m$, with corresponding (x, y, z, a, b) given by

$$(n^2 - 1,\ 2mn,\ 2n,\ m^2 + n^2,\ n^2 + 1).$$

The solutions of (3.11) are well known, corresponding to $m + n\sqrt{2}$ being an odd power of the fundamental unit $1 + \sqrt{2}$ in the ring $\mathbb{Z}[\sqrt{2}]$: namely $m = p_i$, where $p_i = 6p_{i-1} - p_{i-2}$, $i \geqslant 2$, and $p_0 = 1$, $p_1 = 7$. This gives numerical values $M = \boxed{7, 41, 239, \ldots}$

3.5. Case V

We demand $p^2 - q^2 = r^2 - s^2$ and put $p + q = K(r - s)$, $K(p - q) = r + s$. Eliminating r, s,

$$M(K^4 - 1)p^2 - 2(M + 2K^2 + MK^4)pq + M(K^4 - 1)q^2 = 0$$

whose discriminant being square demands that

$$(M + K^2)(MK^2 + 1) = \square.$$

Assuming $M + K^2 = (A + K)^2$, then $M = 2AK + A^2$, so that $2AK^3 + A^2K^2 + 1 = \square = (A\rho - 1)^2$, say, giving $A = 2(\rho + K^3)/(\rho^2 - K^2)$. If we choose $\rho = K^2$ there results $K = (A + 2)/A$, $M = A^2 + 2A + 4$, so that setting $w = A + 1$, we have

$$M = w^2 + 3, \qquad (p, q, r, s) = (w(3 + w^2),\ -2(1 + w^2),\ -w(1 + w^2),\ 2),$$

with point on (1.6)

$$(X, Y, Z) = \Big((-1+w)(2-w+w^2),\ (1+w)(2+w+w^2),\ 2(3+w^2)(4+w^2+2w^4+w^6)\Big).$$

Numerical values are $\boxed{M = 12, 19, 28, 39, 52, 67, 84, 103, 124, 147, 172, 199, 228, \ldots}$

Choosing instead $\rho = K + 2$, then

$$\begin{aligned} A = (K^2 - K + 2)/2, \quad M &= (K + 1)(K + 2)(K^2 - K + 2)/4, \\ (p, q, r, s) &= \Big((K + 1)^2(K^2 - K + 2),\ (K - 1)(K + 2)(K^2 + 1), \\ &\qquad 2(K + 1)(K^2 + 1),\ 2(K - 1)\Big), \end{aligned}$$

with point on (1.6)

$$\begin{aligned} (X, Y, Z) = &\Big(2K(K^3 + K^2 + 2),\ 2(K^2 + K + 2), \\ &\quad 2(K + 1)(K + 2)(K^2 - K + 2)(K^6 + 2K^5 + 3K^4 + 4K^3 + 4K^2 + 2)\Big). \end{aligned}$$

Numerical values are $\boxed{M = 7, 12, 33, 40, 96, 105, 220, \ldots}$

Finally, if we set $2AK^3 + A^2K^2 + 1 = (AK + K^2 - 1)^2$, then $A = K(K^2 - 2)/2$ and $M = K^2(K^4 - 4)/4$, with corresponding $(X, Y, Z) = (2(K^4 - 2), 2K^2, 2(K^2 - 2)(K^2 + 2)(K^8 - 2K^4 + 2))$. On setting $K^2 = 2w$ this gives rise to the parametrization

$$M = 2w(w^2 - 1), \quad (X, Y, Z) = (2w^2 - 1, w, (w^2 - 1)(8w^4 - 4w^2 + 1)).$$

Numerical values are $\boxed{M = 12, 48, 120, 240, \ldots}$.

The curve with $M = w^2 + 3$ represents a K3 elliptic surface:

$$E : v^2 = u\left(u - (w^2 + 3)^2\right)\left(u - (w^2 + 4)(w^2 + 2)\right).$$

and it is possible to show by the Shioda formula that the rank of E over $\mathbb{C}(w)$ is equal to 1. It is likely that

$$P = \left(\frac{(w^2 - w + 2)^2(w^2 + 3)}{(w - 1)^2}, \frac{(w + 1)(w^2 + 3)(w^2 + w + 2)(w^2 - w + 2)}{(w - 1)^3}\right)$$

is a generator for the group (in which case the field of definition of the group is actually $\mathbb{Q}(w)$), and this could be verified as above using a height argument, although we have not undertaken the computation. As before, therefore, we can determine infinitely many parametrized solutions to the ladder problem by computing multiples of P.

4. Rank data

We list here the rank of the elliptic curve (1.8) in the range $3 \leqslant M < 200$ (computed with the aid of Magma [1]).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0			0	0	0	0	0	1	0	0	1	1	1	0	1	0	0	1	0	1
20	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	0	0	0	1	1
40	1	2	1	0	1	1	0	1	1	0	0	1	2	1	1	0	0	1	1	2
60	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	1
80	1	1	1	1	1	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0
100	1	0	1	3	0	1	0	1	1	1	0	1	1	1	0	2	0	0	1	1
120	1	1	0	0	2	0	1	1	0	0	0	2	1	1	0	1	1	2	1	1
140	1	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	1	1	1	1
160	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	2
180	2	1	1	1	0	0	1	1	1	0	0	0	0	0	2	1	1	0	1	2

There are 111 instances in the range $1 < M < 200$ of curves with positive rank, of which we have identified 39 as coming from parametrized families. Of course it is unlikely that every curve of positive rank arises from a parametrization. For example, the curve at $M = 127$ has rank 1 and the smallest solution of the equation at (1.6) is given by $(X, Y, Z) = (59914079, 205805825, 3132229187148973634)$.

References

[1] Bosma, W., Cannon, J., Playoust, C., The Magma algebra system. I. The user language. J. Symbolic Comput., 24(3-4): (1997), 235–265.

[2] Bremner, A., Jones, J.W., On the equation $x^4 + mx^2y^2 + y^4 = z^2$, *J. Number Theory* 50 (1995), no. 2, 286–298.

[3] Brown, E., $x^4 + dx^2y^2 + y^4 = z^2$: Some cases with only trivial solutions - and a solution Euler missed, *Glasgow Math. J.*, 31 (1989), 297–307.

[4] Høibakk, R., Jorstad, T., Lukkassen, D., Lystad, L.P., Integer Crossed Ladders; parametric representations and minimal integer values, Normat, *Nordisk Matematisk Tidsskrift*, 56, 2, (2008), 68–79.

[5] Høibakk, R., Lukkassen, D., Crossed Ladders and Power Means, *Elem. Math.*, 63, 3, (2008), 137–140.

[6] Kuwata, M., The canonical height and elliptic surfaces, J. Number Theory 36 no. 2, (1990), 201–211.

[7] Pocklington,H.C., Some Diophantine impossibilities, *Proc. Cambridge Philos. Soc.* 17 (1914), 108–121.

[8] Shioda, T., On the Mordell-Weil lattices, *Comment. Math. Univ. St. Paul.* 39 no. 2, (1990), 211–240.

[9] Singmaster, D.B., Sources in Recreational Mathematics, *An Annotated Bibliography, Eighth Preliminary Edition,* (2004), http://www.gotham-corp.com/sources.htm

[10] Sinha, T.N., A class of quartic Diophantine equations with only trivial solutions, *Amer. J. Math.*, 100 (1978), 585–590.

[11] Zhang, M.Z., On the Diophantine equation $x^4 + kx^2y^2 + y^4 = z^2$, *Sichuan Daxue Xuebao*, 2 (1983), 24–31.

A. Bremner
School of Mathematics, Arizona State University, Tempe AZ 85287, USA
e-mail: bremner@asu.edu

R. Høibakk
Narvik University College, P.O.B. 385, N-8505 Narvik, Norway
e-mail: rh@hin.no

D. Lukkassen
Narvik University College, P.O.B. 385, N-8505 Narvik, Norway

and

Norut Narvik, P.O.B. 250, N-8504 Narvik, Norway
e-mail: dl@hin.no

Annales Mathematicae et Informaticae
36 *(2009) pp. 43–46*
http://ami.ektf.hu

On the existence of triangle with given angle and opposite angle bisectors length*

József Bukor

Department of Mathematics
J. Selye University, Komárno, Slovakia

Submitted 21 June 2009; Accepted 30 September 2009

Abstract

Let us denote by l_a, l_c the lengths of angle bisectors on the sides BC and AB, respectively. We prove that for given positive $l_a,\ l_c$ and angle $\beta = ABC\angle$ there is a unique triangle.

Keywords: triangle, bisector

MSC: 51-99

It is known that for given lengths of three angle bisectors there is always a unique triangle [2], for an elementary proof, see [5]. In this note we consider the question of existence of a triangle with given angle and the lengts of two angle bisectors. The proposed question was motivated by the works of V. Oxman [3, 4], where the conditions for the existence of a triangle with given length of one side and two angle bisectors were studied.

Using methods of elementary calculus we prove the following theorem

Theorem. *Given positive $l_a,\ l_c,\ \beta < \pi$, there is a unique triangle ABC with $\beta = ABC\angle$ and lengths of bisectors of angles to the sides BC, AB equal to $l_a,\ l_c$.*

Proof. Recall that that in a triangle ABC with sidelengths a, b, c the bisector of angle $CAB\angle$ has length

$$l_a = \sqrt{bc\left(1 - \frac{a^2}{(b+c)^2}\right)}. \tag{1}$$

We shall prove that for given $BC = 1$, $\beta = ABC\angle$ and $p = \frac{l_a}{l_c}$ there is a unique triangle. For simplicity, let us denote the side lengths AB and AC by x and y, respectively. See Figure 1.

*Supported by KEGA grant no. 3/5277/07.

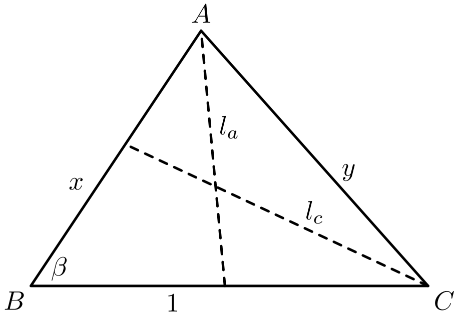


Figure 1

By the well-known Steiner-Lehmus theorem if a triangle has two equal bisectors ($p = 1$), then it is an isoscales triangle. Without lost of generality we may suppose $p > 1$.

From (1) we have

$$l_a^2 = xy\left(1 - \frac{1}{(x+y)^2}\right) \quad \text{and} \quad l_c^2 = y\left(1 - \frac{x^2}{(y+1)^2}\right).$$

Therefore

$$p^2 = \frac{l_a^2}{l_c^2} = \frac{x(x+y-1)(y+1)^2}{(x+y)^2(y+1-x)}.$$

Let us consider the function

$$f(x) = \frac{x(x+y-1)(y+1)^2}{(x+y)^2(y+1-x)}.$$

Note, y is a function of x, since by the law of cosines

$$y = \sqrt{x^2 + 1 - 2x\cos\beta}.$$

For convenience we ignore the dependence of y on x in notation. Our goal is to show that the equation $f(x) = p^2$ has a unique solution.

By the stronger form of Steiner-Lehmus theorem (see, e.g. [1])

$$l_a > l_c \iff a < c,$$

we immediately have that $x > 1$.
Obviously, $f(x)$ is a continuous function on the interval $[1, \infty)$. It is easy to check that

$$\lim_{x\to 1} f(x) = 1 \quad \text{and} \quad \lim_{x\to\infty} f(x) = \infty.$$

By the above and the continuity of $f(x)$, Bolzano's theorem implies the existence of a solution of $f(x) = p^2$.

To prove the uniqueness we show that the function $f(x)$ is strictly increasing on $[1, \infty)$.
Since the derivative of the function y

$$y' = \frac{x - \cos\beta}{\sqrt{x^2 + 1 - 2x\cos\beta}}$$

is positive on $[1, \infty)$, hence y is strictly increasing throughout that interval.
Then

$$\frac{x + y - 1}{x + y} = 1 - \frac{1}{x + y} \tag{2}$$

is strictly increasing on $[1, \infty)$, too. Since

$$(y + 1 - x)' = y' - 1 = \frac{\cos^2\beta - 1}{(x - \cos\beta + \sqrt{x^2 + 1 - 2x\cos\beta})\sqrt{x^2 + 1 - 2x\cos\beta}}$$

is negative, we deduce that

$$\frac{1}{y + 1 - x} \tag{3}$$

strictly increases for $x \geqslant 1$.
Let

$$g(x) = \ln\frac{x(y + 1)^2}{x + y}.$$

Then

$$g'(x) = \frac{1}{x} + \frac{2y'}{y + 1} - \frac{1 + y'}{x + y}$$

which can be rewritten into the form

$$g'(x) = \big(y^2 + y + xyy' + xy'(2x - 1)\big)\,\frac{1}{x(y + 1)(x + y)}.$$

Clearly, $g'(x)$ is positive for any $x \geqslant 1$. From this follows that

$$\frac{x(y + 1)^2}{x + y} \tag{4}$$

is strictly increasing on $[1, \infty)$ (the positive function is strictly increasing if and only if its natural logarithm is strictly increasing). Taking into account that $f(x)$ is a product of functions (2–4) which strictly increase on the interval $[1, \infty)$, the assertion follows.

We have actually proved that for given $BC = 1$, $\beta = ABC\angle$ and $p = \frac{l_a}{l_c}$ there is a unique triangle. If two triangles are similar then their corresponding angle bisectors are proportionate. Using the similarity of triangles it can be easily deduced that for given l_a, l_c, β there is a unique triangle if and only if there exists a triangle for given $BC = 1$, β and $p = \frac{l_a}{l_c}$. □

References

[1] Coxeter, H.S.M., Greitzer, S.L., Geometry revisited, New York, *Random House*, 1967.

[2] Mironescu, P., Panaitopol, L., The existence of a triangle with prescribed angle bisector lengths, *Amer. Math. Monthly*, 101 (1994) 58–60.

[3] Oxman, V., On the existence of triangles with given lengths of one side and two adjacent angle bisectors, *Forum Geom.*, 4 (2004) 215–218.

[4] Oxman, V., On the existence of triangles with given lengths of one side, the opposite and one adjacent angle bisectors, *Forum Geom.*, 5 (2005) 21–22.

[5] Zhukov, A., Akulich, I., Is the triangle defined uniquely? (Odnoznachno li opredelyaetsya treugolnik?) (Russian), *Kvant*, No. 1 (2003) 29–31.

József Bukor
Department of Mathematics
J. Selye University
P.O.Box 54
945 01 Komárno
Slovakia
e-mail: bukorj@selyeuni.sk

Annales Mathematicae et Informaticae
***36** (2009) pp. 47–60*
http://ami.ektf.hu

Inclusion properties of the intersection convolution of relations

Judita Dascăl, Árpád Száz

Institute of Mathematics, University of Debrecen

Submitted 9 November 2008; Accepted 25 February 2009

Abstract

For various relations F and G on one groupoid X with zero to another Y, we establish several simple, but important inclusions among the relations F, G, $F*G$, $F+G(0)$, and $F(0)+G$.

The latter relations are given here by $\big(F+G(0)\big)(x)=F(x)+G(0)$, $\big(F(0)+G\big)(x)=F(0)+G(x)$, and

$$\big(F*G\big)(x)=\bigcap\big\{F(u)+G(v):x=u+v,F(u)\neq\emptyset,G(v)\neq\emptyset\big\}$$

for all $x\in X$. The intersection convolution $*$ allows of a natural generalization of the Hahn-Banach type extension theorems.

Keywords: Groupoids, binary relations, intersection convolution.

MSC: Primary 20L13; Secondary 46A22.

1. A few basic facts on relations and groupoids

A subset F of a product set $X\times Y$ is called a relation on X to Y. For each $x\in X$, the set $F(x)=\{y\in X:(x,y)\in F\}$ is called the image of x under F, or the value of F at x.

Now, the set $D_F=\{x\in X:F(x)\neq\emptyset\}$ may be naturally called the domain of F. Moreover, if in particular $D_F=X$, then we may say that F is a relation of X to Y, or that F is a total relation on X to Y.

In particular, a relation f on X to Y is called a function if for each $x\in D_f$ there exists $y\in Y$ such that $f(x)=\{y\}$. In this case, by identifying singletons with their elements, we may simply write $f(x)=y$.

If X is a set and $+$ is a function of X^2 to X, then the function $+$ is called an operation in X and the ordered pair $X(+)=(X,+)$ is called a groupoid even if X is void.

In this case, we may simply write $x + y$ in place of $+(x, y)$ for any $x, y \in X$. Moreover, we may also simply write X in place of $X(+)$ whenever the operation $+$ is clearly understood.

In practical applications, instead of groupoids, it is usually sufficient to consider only semigroups. However, several definitions and theorems on semigroups can be naturally extended to groupoids.

For instance, if X is a groupoid, then for any $A, B \subset X$, we may naturally write $A+B = \{a+b : a \in A, b \in B\}$. Moreover, we may also write $x+A = \{x\}+A$ and $A+x = A+\{x\}$ for any $x \in X$.

Note that if in particular X is a group, then we may also naturally write $-A = \{-a : a \in A\}$ and $A - B = A + (-B)$ for any $A, B \subset X$. Though, the family $\mathcal{P}(X)$ of all subsets of X is only a semigroup with zero.

Now, if F and G are relations on a set X to a groupoid Y, then the pointwise sum $F+G$ of F and G can be naturally defined such that $(F+G)(x) = F(x)+G(x)$ for all $x \in X$.

Note that if in particular X is also a groupoid, then the above pointwise sum of the relations F and G may be easily confused with the global sum $F \oplus G = \big\{(x+z, y+w) : (x, y) \in F, (z, w) \in G\big\}$.

2. The most important additivity properties of relations

Analogously to the usual definition of superadditive functions, we may naturally consider the following

Definition 2.1. A relation F on one groupoid X to another Y is called superadditive if for any $x, y \in X$ we have

$$F(x) + F(y) \subset F(x+y).$$

Remark 2.2. Note that thus F is superadditive if and only if $F \oplus F \subset F$. That is, F is a subgroupoid of $X \times Y$.

Moreover, if in particular F is a reflexive, superadditive relation of X to itself, then F is already a translation relation in the sense that $x + F(y) \subset F(x+y)$ for all $x, y \in X$.

In addition to Definition 2.1, we may also naturally introduce the following

Definition 2.3. A relation F on one groupoid X to another Y is called

(1) subadditive if $F(x+y) \subset F(x) + F(y)$ for all $x, y \in X$;

(2) semi-subadditive if $F(x+y) \subset F(x) + F(y)$ for all $x, y \in D_F$;

(3) quasi-subadditive if $F(x+y) \subset F(x) + F(y)$ for all $x, y \in X$ with either $x \in D_F$ or $y \in D_F$.

Remark 2.4. Now, the relation F may, for instance, be naturally called quasi-additive if it is both superadditive and quasi-subadditive.

In [9], by calling a relation F on one group X to another Y quasi-odd if $-F(x) \cap F(-x) \neq \emptyset$ for all $x \in D_F$, the second author has shown that a nonvoid, quasi-odd, superadditive relation is already quasi-additive.

As some obvious generalizations of the above definitions, we may also naturally introduce the following definitions.

Definition 2.5. A relation F on a groupoid X with zero to an arbitrary groupoid Y is called

(1) zero-superadditive if $F(x) + F(0) \subset F(x)$ and $F(0) + F(x) \subset F(x)$ for all $x \in X$;

(2) zero-subadditive if $F(x) \subset F(x) + F(0)$ and $F(x) \subset F(0) + F(x)$ for all $x \in X$.

Definition 2.6. A relation F on a group X to a groupoid Y is called

(1) inversion-superadditive if $F(x) + F(-x) \subset F(0)$ for all $x \in X$;

(2) inversion-subadditive if $F(0) \subset F(x) + F(-x)$ for all $x \in X$;

(3) inversion-quasi-subadditive if $F(0) \subset F(x) + F(-x)$ for all $x \in D_F$.

Remark 2.7. Note that, in the latter case, we also have $F(0) \subset F(-x) + F(x)$ for all $x \in D_F$.

Namely, if $F(0) \neq \emptyset$, then by the inversion-quasi-subadditivity of F we also have $F(-x) \neq \emptyset$, and thus $-x \in D_F$ for all $x \in D_F$.

3. The intersection convolution of relations

Definition 3.1. If X is a groupoid, then for any $x \in X$ and $A, B \subset X$, we define

$$\Gamma(x, A, B) = \big\{ (u, v) \in A \times B : x = u + v \big\}.$$

Remark 3.2. Now, in particular, we may simply write $\Gamma(x) = \Gamma(x, X, X)$. Thus, Γ is just the inverse relation of the operation $+$ in X. Moreover, we have

$$\Gamma\big(x, A, B\big) = \Gamma(x) \cap (A \times B).$$

Definition 3.3. If F and G are relations on one groupoid X to another Y, then we define a relation $F * G$ on X to Y such that

$$\big(F * G\big)(x) = \bigcap \big\{ F(u) + G(v) : (u, v) \in \Gamma(x, D_F, D_G) \big\}$$

for all $x \in X$. The relation $F * G$ is called the intersection convolution of the relations F and G.

Remark 3.4. If in particular F and G are relations of X to Y, then we may simply write

$$\big(F*G\big)(x)=\bigcap_{x=u+v}\big(F(u)+G(v)\big)=\bigcap\big\{F(u)+G(v):(u,v)\in\Gamma(x)\big\}.$$

A particular case of Definition 3.3 was already considered in [6]. But, the following theorems have only been proved in [9].

Theorem 3.5. *If F, G, H, and K are relations on one groupoid X to another Y such that*

(1) $D_H\subset D_F$ and $F(u)\subset H(u)$ for all $u\in D_H$;

(2) $D_K\subset D_G$ and $G(v)\subset K(v)$ for all $v\in D_K$;

then $F*G\subset H*K$.

Now, as some immediate consequences of this theorem, we can also state

Corollary 3.6. *If F, G, and H are relations on one groupoid X to another Y such that $D_H\subset D_F$ and $F(u)\subset H(u)$ for all $u\in D_H$, then $F*G\subset H*G$.*

Corollary 3.7. *If F, G, and H are relations on one groupoid X to another Y such that $D_H\subset D_G$ and $G(v)\subset H(v)$ for all $v\in D_H$, then $F*G\subset F*H$.*

Theorem 3.8. *If F and G are relations on a group X to a groupoid Y, then for any $x\in X$ we have*

$$(F*G)(x)=\bigcap\big\{F(x-v)+G(v):v\in(-D_F+x)\cap D_G\big\}=$$
$$=\bigcap\big\{F(u)+G(-u+x):u\in D_F\cap(x-D_G)\big\}.$$

Hence, by using that $-X+x=X$ and $x-X=X$ for all $x\in X$, we can immediately get

Corollary 3.9. *If F and G are relations on a group X to a groupoid Y, then for any $x\in X$ we have*

(1) $\big(F*G\big)(x)=\bigcap_{v\in D_G}\big(F(x-v)+G(v)\big)$ whenever F is total;

(2) $\big(F*G\big)(x)=\bigcap_{u\in D_F}\big(F(u)+G(-u+x)\big)$ whenever G is total.

Hence, it is clear that in particular we also have

Corollary 3.10. *If F and G are relations of a group X to a groupoid Y, then for any $x\in X$ we have*

$$(F*G)(x)=\bigcap_{v\in X}\big(F(x-v)+G(v)\big)=\bigcap_{u\in X}\big(F(u)+G(-u+x)\big).$$

4. Convolutional inclusions for quite general relations

By using the corresponding definitions, we can easily prove the following

Theorem 4.1. *If F and G are relations on a groupoid X with zero to an arbitrary one Y, then*

(1) $(F*G)(x) \subset \big(F+G(0)\big)(x)$ for all $x \in D_F$ if $G(0) \neq \emptyset$;

(2) $(F*G)(x) \subset \big(F(0)+G\big)(x)$ for all $x \in D_G$ if $F(0) \neq \emptyset$.

Proof. If $x \in D_F$ and $G(0) \neq \emptyset$, then $(x,0) \in \Gamma(x, D_F, D_G)$. Therefore,

$$\begin{aligned}\big(F*G\big)(x) = & \bigcap \big\{ F(u)+G(v) : (u,v) \in \Gamma(x, D_F, D_G) \big\} \subset \\ & \subset F(x) + G(0) = \big(F+G(0)\big)(x). \end{aligned} \qquad \square$$

In addition to this theorem, it is also worth proving the following two theorems.

Theorem 4.2. *If F and G are relations of one groupoid X with zero to another Y, then*

(1) $F \subset F+G(0)$ if $0 \in G(0)$;

(2) $G \subset F(0)+G$ if $0 \in F(0)$.

Proof. If the condition of (1) holds, then

$$F(x) = F(x) + \{0\} \subset F(x) + G(0) = \big(F+G(0)\big)(x)$$

for all $x \in X$. Therefore, the conclusion of (1) also holds. □

Theorem 4.3. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F+G(0) \subset F$ if $G(0) \subset \{0\}$;

(2) $F(0)+G \subset G$ if $F(0) \subset \{0\}$.

Proof. If the condition of (1) holds, then

$$\big(F+G(0)\big)(x) = F(x) + G(0) \subset F(x) + \{0\} = F(x)$$

for all $x \in X$. Therefore, the conclusion of (1) also holds. □

Now, as an immediate consequence of the latter two theorems, we can also state

Corollary 4.4. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F = F+G(0)$ if $G(0) = \{0\}$;

(2) $G = F(0)+G$ if $F(0) = \{0\}$.

Moreover, as an immediate consequence of Theorems 4.1 and 4.3, we can also state

Theorem 4.5. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $(F * G)(x) \subset F(x)$ for all $x \in D_F$ if $G(0) = \{0\}$;

(2) $(F * G)(x) \subset G(x)$ for all $x \in D_G$ if $F(0) = \{0\}$.

Hence, it is clear that in particular we also have

Corollary 4.6. *If F is a relation on one groupoid X with zero to another Y such that $F(0) = \{0\}$, then $(F * F)(x) \subset F(x)$ for all $x \in D_F$.*

5. Inclusions for zero-subadditive and zero-superadditive relations

In addition to Theorems 4.2 and 4.3, we can also easily prove the following two theorems.

Theorem 5.1. *If F and G are relations on a groupoid X with zero to an arbitrary one Y, then*

(1) $F \subset F + G(0)$ if F is zero-subadditive and $F(0) \subset G(0)$;

(2) $G \subset F(0) + G$ if G is zero-subadditive and $G(0) \subset F(0)$.

Proof. If the conditions of (1) hold, then

$$F(x) \subset F(x) + F(0) \subset F(x) + G(0) = \big(F + G(0)\big)(x)$$

for all $x \in X$. Therefore, the conclusion of (1) also holds. □

Theorem 5.2. *If F and G are relations on a groupoid X with zero to an arbitrary one Y, then*

(1) $F + G(0) \subset F$ if F is zero-superadditive and $G(0) \subset F(0)$;

(2) $F(0) + G \subset G$ if G is zero-superadditive and $F(0) \subset G(0)$.

Proof. If the conditions of (1) hold, then

$$\big(F + G(0)\big)(x) = F(x) + G(0) \subset F(x) + F(0) \subset F(x)$$

for all $x \in X$. Therefore, the conclusion of (1) also holds. □

Now, as an immediate consequence of the latter two theorems, we can also state

Corollary 5.3. *If F and G are relations on a groupoid X with zero to an arbitrary one Y, then*

(1) $F = F + G(0)$ if F is zero-additive and $F(0) = G(0)$;

(2) $G = F(0) + G$ if G is zero-additive and $G(0) = F(0)$.

Moreover, combining Theorems 4.3 and 4.2 with Theorems 5.1 and 5.2, respectively, we can also at once state the following two theorems.

Theorem 5.4. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F = F + G(0)$ if F is zero-subadditive and $F(0) \subset G(0) \subset \{0\}$;

(2) $G = F(0) + G$ if G is zero-subadditive and $G(0) \subset F(0) \subset \{0\}$.

Theorem 5.5. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F = F + G(0)$ if F is zero-superadditive and $0 \in G(0) \subset F(0)$;

(2) $G = F(0) + G$ if G is zero-superadditive and $0 \in F(0) \subset G(0)$.

On the other hand, as an immediate consequence of Theorems 4.1 and 5.2, we can also state

Theorem 5.6. *If F and G are relations on a groupoid X with zero to an arbitrary one Y, then*

(1) $(F * G)(x) \subset F(x)$ for all $x \in D_F$ if F is zero-superadditive and $\emptyset \neq G(0) \subset F(0)$;

(2) $(F * G)(x) \subset G(x)$ for all $x \in D_G$ if G is zero-superadditive and $\emptyset \neq F(0) \subset G(0)$.

Hence, it is clear that in particular we also have

Corollary 5.7. *If F is a zero-superadditive relation on a groupoid X with zero to an arbitrary one Y such that $F(0) \neq \emptyset$, then $(F * F)(x) \subset F(x)$ for all $x \in D_F$.*

6. Convolutional inclusions for superadditive and semi-subadditive relations

In addition to Theorem 5.6, it is also worth proving the following.

Theorem 6.1. *If F, G, and H are relations on one groupoid X to another Y and $x \in D_F + D_G$ such that*

$$F(u) + G(v) \subset H(u+v)$$

for any $u \in D_F$ and $v \in D_G$ with $x = u + v$, then

$$(F * G)(x) \subset H(x).$$

Proof. By the above assumptions, it is clear that

$$\begin{aligned}\big(F * G\big)(x) &= \bigcap \big\{ F(u) + G(v) : (u, v) \in \Gamma(x, D_F, D_G) \big\} \subset \\ &\subset \bigcap \big\{ H(u+v) : (u, v) \in \Gamma(x, D_F, D_G) \big\} = \\ &= \bigcap \big\{ H(x) : (u, v) \in \Gamma(x, D_F, D_G) \big\} = H(x).\end{aligned}$$

□

Now, as an immediate consequence of this theorem, we can also state

Corollary 6.2. *If F and G are relations on one groupoid X to another Y and $x \in D_F + D_G$, then*

(1) $(F * G)(x) \subset F(x)$ if $F(u) + G(v) \subset F(u+v)$ for any $u \in D_F$ and $v \in D_G$ with $x = u + v$;

(2) $(F * G) \subset G(x)$ if $F(u) + G(v) \subset G(u+v)$ for any $u \in D_F$ and $v \in D_G$ with $x = u + v$.

Hence, it is clear that in particular we also have

Corollary 6.3. *If F is a superadditive relation on one groupoid X to another Y, then $(F * F)(x) \subset F(x)$ for all $x \in D_F + D_F$.*

Analogously to Theorem 6.1, we can also easily prove the following

Theorem 6.4. *If F, G, and H are relations on one groupoid X to another Y and $x \in D_G + D_H$ such that*

$$F(u+v) = G(u) + H(v)$$

for any $u \in D_G$ and $v \in D_H$ with $x = u + v$, then

$$F(x) = (G * H)(x).$$

Now, as an immediate consequence of this theorem, we can also state

Corollary 6.5. *If F and G are relations on one groupoid X to another Y and $x \in D_F + D_G$, then*

(1) $F(x) = (F * G)(x)$ if $F(u+v) = F(u) + G(v)$ for any $u \in D_F$ and $v \in D_G$ with $x = u + v$;

(2) $G(x) = (F * G)(x)$ if $G(u+v) = F(u) + G(v)$ for any $u \in D_F$ and $v \in D_G$ with $x = u + v$.

Hence, it is clear that in particular we also have

Corollary 6.6. *If F is a semi-additive relation on one groupoid X to another Y, then $F(x) = (F * F)(x)$ for all $x \in D_F + D_F$.*

Moreover, as a counterpart of Theorem 6.1, we can also prove the following

Theorem 6.7. *If F, G, and H are relations on one groupoid X to another Y, then for any $x \in X$ the following assertions are equivalent:*

(1) $F(x) \subset (G * H)(x)$;

(2) $F(u+v) \subset G(u) + H(v)$ for any $u \in D_G$ and $v \in D_H$ with $x = u + v$.

Proof. If (1) holds and $u \in D_G$ and $v \in D_H$ such that $x = u + v$, then

$$\begin{aligned} F(u+v) = F(x) \subset (G * H)(x) = \\ = \bigcap \big\{ G(s) + H(t) : (s,t) \in \Gamma(x, D_G, D_H) \big\} \subset G(u) + H(v) \end{aligned}$$

since $(u,v) \in \Gamma(x, D_G, D_H) \big\}$. Thus, (2) also holds.

While, if (2) holds, then for any $(u,v) \in \Gamma(x, D_G, D_H)$ we have

$$F(x) = F(u+v) \subset G(u) + H(v)$$

since $u \in D_G$ and $v \in D_H$ such that $x = u + v$. Hence, it is clear that

$$F(x) \subset \bigcap \big\{ G(u) + H(v) : (u,v) \in \Gamma(x, D_G, D_H) \big\} = (G * H)(x).$$

Therefore, (1) also holds. □

Now, as an immediate consequence of this theorem, we can also state

Corollary 6.8. *If F and G are relations on one groupoid X to another Y, then for any $x \in X$ we have:*

(1) $F(x) \subset (F * G)(x)$ if and only if $F(u+v) \subset F(u) + G(v)$ for any $u \in D_F$ and $v \in D_G$ with $x = u + v$;

(2) $G(x) \subset (F * G)(x)$ if and only if $G(u+v) \subset F(u) + G(v)$ for any $u \in D_F$ and $v \in D_G$ with $x = u + v$.

Hence, it is clear that in particular we also have

Corollary 6.9. *If F is a relation on one groupoid X to another Y, then for any $x \in X$ the following assertions are equivalent:*
(1) $F(x) \subset (F * F)(x)$;
(2) $F(u+v) \subset F(u) + F(v)$ for any $u, v \in D_F$ with $x = u + v$.

7. Convolutional equalities for semi-subadditive and zero-superadditive relations

Now, as a useful characterization of semi-subadditivity, we can also state

Theorem 7.1. *If F is a relation on one groupoid X to another Y, then the following assertions are equivalent:*
(1) $F \subset F * F$;
(2) F is semi-subadditive.

Proof. If (1) holds, then in particular for any $u, v \in D_F$, we have $F(u+v) \subset (F * F)(u+v)$. Hence, by using Corollary 6.9, we can infer that $F(u+v) \subset F(u) + F(v)$. Therefore, (2) also holds.

Conversely, if (2) holds and $x \in X$, then in particular for any $u, v \in D_F$, with $x = u + v$, we have $F(u+v) \subset F(u) + F(v)$. Hence, by using Corollary 6.9, we can infer that $F(x) \subset (F * F)(x)$. Therefore, (1) also holds. □

From this theorem, by using Corollaries 3.6 and 3.7, we can immediately derive

Corollary 7.2. *If F and G are relations on one groupoid X to another Y, then*

(1) $F \subset F * G$ if F is semi-subadditive, $D_G \subset D_F$, and $F(x) \subset G(x)$ for all $x \in D_G$;

(2) $G \subset F * G$ if G is semi-subadditive, $D_F \subset D_G$, and $G(x) \subset F(x)$ for all $x \in D_F$.

Now, as an immediate consequence of Theorem 4.5 and Corollary 7.2, we can also state

Theorem 7.3. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F = F * G$ if F is total and subadditive, $F(x) \subset G(x)$ for all $x \in D_G$, and $G(0) = \{0\}$;

(2) $G = F * G$ if G is total and subadditive, $G(x) \subset F(x)$ for all $x \in D_F$, and $F(0) = \{0\}$.

Hence, it is clear that in particular we also have

Corollary 7.4. *If F is a subadditive relation of one groupoid X with zero to another Y such that $F(0) = \{0\}$, then $F = F * F$.*

Moreover, as an immediate consequence of Theorem 5.6 and Corollary 7.2, we can also state

Theorem 7.5. *If F and G are relations of a groupoid X with zero to an arbitrary one Y, then*

(1) $F = F * G$ if F is total, subadditive, and zero-superadditive, $F(x) \subset G(x)$ for all $x \in D_G$, and $\emptyset \neq G(0) \subset F(0)$;

(2) $G = F * G$ if G is total, subadditive, and zero-superadditive, $G(x) \subset F(x)$ for all $x \in D_F$, and $\emptyset \neq F(0) \subset G(0)$.

Hence, it is clear that in particular we also have

Corollary 7.6. *If F is a subadditive and zero-superadditve relation of a groupoid X with zero to an arbitrary one Y, then $F = F * F$.*

On the other hand, from Corollary 6.5, we can immediately get

Theorem 7.7. *If F and G are relations on one groupoid X to another Y such that $X = D_F + D_G$, then*

(1) $F = F * G$ if $F(u + v) = F(u) + G(v)$ for all $u \in D_F$ and $v \in D_G$;

(2) $G = F * G$ if $G(u + v) = F(u) + G(v)$ for all $u \in D_F$ and $v \in D_G$.

Hence, it is clear that in particular we also have

Corollary 7.8. *If F is a semi-additive relation on one groupoid X to another Y such that $X = D_F + D_F$, then $F = F * F$.*

8. Convolutional inclusions for zero-subadditive and zero-superadditive relations

From Theorem 4.2, by using Corollaries 3.6 and 3.7, we can immediately get

Theorem 8.1. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F*G \subset \big(F+G(0)\big)*G$ *if* $0 \in G(0)$*;*

(2) $F*G \subset F*\big(F(0)+G\big)$ *if* $0 \in F(0)$*.*

Moreover, as an immediate consequence of Corollary 4.4, we can also state

Theorem 8.2. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F*G = \big(F+G(0)\big)*G$ *if* $G(0)=\{0\}$*;*

(2) $F*G = F*\big(F(0)+G\big)$ *if* $F(0)=\{0\}$*.*

On the other hand, from Theorems 5.1 and 5.2, by using Corollaries 3.6 and 3.7, we can immediately get the following theorems.

Theorem 8.3. *If F and G are relations on a groupoid X with zero to an arbitrary one Y, then*

(1) $F*G \subset \big(F+G(0)\big)*G$ *if F is zero-subadditive and* $F(0) \subset G(0)$*;*

(2) $F*G \subset F*\big(F(0)+G\big)$ *if G is zero-subadditive and* $G(0) \subset F(0)$*.*

Theorem 8.4. *If F and G are relations of a groupoid X with zero to an arbitrary one Y, then*

(1) $\big(F+G(0)\big)*G \subset F*G$ *if F is zero-superadditive and* $\emptyset \neq G(0) \subset F(0)$*;*

(2) $F*\big(F(0)+G\big) \subset F*G$ *if G is zero-superadditive and* $\emptyset \neq F(0) \subset G(0)$*.*

Now, as an immediate consequence of these theorems, we can also state

Corollary 8.5. *If F and G are relations on a groupoid X with zero to an arbitrary one Y such that $F(0)=G(0)\neq\emptyset$, then*

(1) $F*G = \big(F+G(0)\big)*G$ *if F is zero-additive;*

(2) $F*G = F*\big(F(0)+G\big)$ *if G is zero-additive.*

Moreover, as some immediate consequences of Theorems 5.4 and 5.5, we can also state

Theorem 8.6. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F*G = \big(F+G(0)\big)*G$ *if F is zero-subadditive and* $F(0) \subset G(0) \subset \{0\}$*;*

(2) $F*G = F*\big(F(0)+G\big)$ *if G is zero-subadditive and* $G(0) \subset F(0) \subset \{0\}$*.*

Theorem 8.7. *If F and G are relations on one groupoid X with zero to another Y, then*

(1) $F * G = \big(F + G(0)\big) * G$ if F is zero-superadditive and $0 \in G(0) \subset F(0)$;

(2) $F * G = F * \big(F(0) + G\big)$ if G is zero-superadditive and $0 \in F(0) \subset G(0)$.

9. Convolutional inclusions for zero-additive and inversion-additive relations

In addition to Theorem 8.3, we can also prove the following

Theorem 9.1. *If F and G are relations on a groupoid X with zero to a semigroup Y, then*

(1) $F * G \subset \big(F + G(0)\big) * G$ if G is zero-subadditive;

(2) $F * G \subset F * \big(F(0) + G\big)$ if F is zero-subadditive.

Proof. If the condition of (1) holds, then

$$F(u) + G(v) \subset F(u) + G(0) + G(v) = \big(F + G(0)\big)(u) + G(v)$$

for all $u, v \in X$. Therefore, for any $x \in X$, we have

$$\begin{aligned}\big(F * G\big)(x) &= \bigcap \big\{ F(u) + G(v) : (u, v) \in \Gamma(x, D_F, D_G) \big\} \subset \\ &\subset \bigcap \big\{ \big(F + G(0)\big)(u) + G(v) : (u, v) \in \Gamma(x, D_F, D_G) \big\} = \\ &= \bigcap \big\{ \big(F + G(0)\big)(u) + G(v) : (u, v) \in \Gamma(x, D_{F+G(0)}, D_G) \big\} = \\ &= \big(\big(F + G(0)\big) * G\big)(x)\end{aligned}$$

provided that $G(0) \neq \emptyset$. Therefore, the conclusion of (1) also holds. Namely, if $G(0) = \emptyset$, then $\big(F + G(0)\big) * G = \emptyset * G = X \times Y$. □

Note that if G is zero-superadditive and $G(0) \neq \emptyset$, then just the converse inclusion holds. Therefore, we can also state the following

Theorem 9.2. *If F and G are relations on a groupoid X with zero to a semigroup Y, then*

(1) $\big(F + G(0)\big) * G \subset F * G$ if G is zero-superadditive and $G(0) \neq \emptyset$;

(2) $F * \big(F(0) + G\big) \subset F * G$ if F is zero-superadditive and $F(0) \neq \emptyset$.

Now, as an immediate consequence of the above theorems, we can also state

Corollary 9.3. *If F and G are relations on a groupoid X with zero to a semigroup Y, then*

(1) $F * G = \big(F + G(0)\big) * G$ if G is zero-additive and $G(0) \neq \emptyset$;

(2) $F * G = F * \big(F(0) + G\big)$ if F is zero-additive and $F(0) \neq \emptyset$.

On the other hand, combining Theorems 8.1 with Theorems 9.2, we can also at once state the following

Theorem 9.4. *If F and G are relations on a groupoid X with zero to a semigroup Y with zero, then*

(1) $F * G = \big(F + G(0)\big) * G$ if G is zero-superadditive and $0 \in G(0)$;

(2) $F * G = F * \big(F(0) + G\big)$ if F is zero-superadditive and $0 \in F(0)$.

Moreover, combining Theorems 8.4 and 8.3 with Theorems 9.1 and 9.2, respectively, we can also at once state the following theorems.

Theorem 9.5. *If F and G are relations on a groupoid X with zero to a semigroup Y, then*

(1) $F * G = \big(F + G(0)\big) * G$ if F is zero-superadditive, G is zero-subadditive, and $\emptyset \neq G(0) \subset F(0)$;

(2) $F * G = F * \big(F(0) + G\big)$ if F is zero-subadditive, G is zero-superadditive, and $\emptyset \neq F(0) \subset G(0)$.

Theorem 9.6. *If F and G are relations on a groupoid X with zero to a semigroup Y, then*

(1) $F * G = \big(F + G(0)\big) * G$ if F is zero-subadditive, G is zero-superadditive, and $F(0) \subset G(0) \neq \emptyset$;

(2) $F * G = F * \big(F(0) + G\big)$ if F is zero-superadditive, G is zero-subadditive, and $G(0) \subset F(0) \neq \emptyset$.

Finally, we note that in addition to Theorem 4.1, we can also prove the following

Theorem 9.7. *If F and G are relations on a group X to a semigroup Y, then*

(1) $F + G(0) \subset F * G$ if F is superadditive, G is inversion-quasi-subadditive and $G \subset F$;

(2) $F(0) + G \subset F * G$ if G is superadditive, F is inversion–quasi-subadditive and $F \subset G$.

Proof. If $x \in X$ and the conditions of (1) hold, then we can easily see that

$$\begin{aligned}\big(F + G(0)\big)(x) = F(x) + G(0) \subset F(x) + G(-v) + G(v) \subset \\ \subset F(x) + F(-v) + G(v) \subset F(x-v) + G(v)\end{aligned}$$

for all $v \in D_G$. Hence, by using Theorem 3.8, we can infer that

$$\big(F + G(0)\big)(x) \subset \bigcap \big\{ F(x-v) + G(v) : v \in (-D_F + x) \cap D_G \big\} = (F * G)(x).$$

Therefore, the conclusion of (1) also holds. □

Now, as an immediate consequence of Theorems 4.1 and 9.7, we can also state

Theorem 9.8. *If F and G are relations on a group X to a semigroup Y, then*

(1) $F * G = F + G(0)$ if F is total and superadditive, G is inversion-quasi-subadditive and $G \subset F$;

(2) $F * G = F(0) + G$ if G is total and superadditive, F is inversion–quasi-subadditive and $F \subset G$.

Hence, it is clear that in particular we also have

Corollary 9.9. *If F is a superadditive and inversion-quasi-subadditive relation of a group X to a semigroup Y, then*

$$F * F = F + F(0) \quad \textit{and} \quad F * F = F(0) + F.$$

References

[1] Beg, I., Fuzzy multivalued functions, *Bull. Allahabad Math. Soc.*, 21 (2006), 41–104.

[2] Boros, Z. and Száz, Á., Reflexivity, transitivity, symmetry, and antisymmetry of the intersection convolution of relations, *Rostock. Math. Kolloq.*, 63 (2008), 55–62.

[3] Czerwik, S., Functional Equations and Inequalities in Several Variables, *World Scientific*, London, 2002.

[4] Glavosits, T. and Száz, Á., Pointwise and global sums and negatives of binary relations, *An. St., Univ. Ovidius Constanta*, 11 (2003), 87–94.

[5] Strömberg, T., The operation of infimal convolution, *Dissertationes Math.*, 352 (1996), 1–58.

[6] Száz, Á., The intersection convolution of relations and the Hahn–Banach type theorems, *Ann. Polon. Math.*, 69 (1998), 235–249.

[7] Száz, Á., Translation relations, the building blocks of compatible relators, *Math. Montisnigri*, 12 (2000), 135–156.

[8] Száz, Á., An extension of an additive selection theorem of Z. Gajda and R. Ger, *Tech. Rep., Inst. Math.*, Univ. Debrecen 8 (2006), 1–24.

[9] Száz, Á., The intersection convolution of relations on one groupoid to another, *Tech. Rep., Inst. Math.*, Univ. Debrecen 2 (2008), 1–22.

[10] Száz, Á., Relationships between the intersection convolution and other important operations on relations, *Math. Pannon.*, 20 (2009), 99–107.

Judita Dascăl
Árpád Száz
Institute of Mathematics
University of Debrecen
H-4010 Debrecen, Pf. 12
Hungary
e-mails: jdascal@math.klte.hu
szaz@math.klte.hu

Annales Mathematicae et Informaticae
36 *(2009) pp. 61–69*
http://ami.ektf.hu

A computational algorithm for the CPP/M/c retrial queue

Tien Van Do

Department of Telecommunications
University of Technology and Economics, Budapest

Submitted 24 November 2008; Accepted 20 April 2009

Abstract

This paper introduces the retrial CPP/M/c queue, which is the generalization of the M/M/c retrial queue. The arrival process of jobs into the queue follows the Compound Poisson Process (CPP). We present an efficient and numerically stable computational algorithm for the steady state probabilities.

Keywords: retrial queue, computational algorithm

MSC: 60-08, 60J22

1. Introduction

Retrial queues have formed one of intensive research topics in the queueing theory [1, 2, 3, 4, 8, 10, 12, 14, 16, 17]. The popularity of retrial queues is explained by the fact that retrial queues can be used to model various problems in real systems such as telecommunication networks, wireless networks and computer systems.

It is well-known that the main M/M/c retrial queue (where the retrial rate depends on the number of customers in the orbit) with $c > 2$ is mathematically untractable. The stationary distributions of the main M/M/c retrial queue with $c > 2$ can be computed using approximation techniques [8]. Falin and Templeton proposed a truncation model and a numerical tractable with a threshold in their book [8].

This paper generalizes the numerical tractable M/M/c retrial queue (where the retrial rate is independent of the number of customers in the orbit). We introduce the retrial CPP/M/c queue with batch arrivals following the Compound Poisson Process (CPP), where the interarrival times have the Generalized Exponential (GE) distribution. Note that the GE is the only distribution of least bias [9], if only the mean and variance are reliably computed from the measurement data. It has been shown in the recent work [7] that the CPP is accurate enough to model Internet traffic (i.e.: CPP parameters were estimated from the captured Internet traffic) and to be used for the performance evaluation in telecommunication systems. We provide a stable computational algorithm for the proposed queue.

In Section 2 we give a description for the CPP/M/c retrial queue. In Section 3 we provide a computational algorithm. In Section 4 we show that our proposed algorithm finds the eigenvalue when the existing approach fails.

2. The CPP/M/c Retrial Queue

Request arrivals follow the CPP with parameter (λ, ω) $(0 \leqslant \omega < 1)$. That is, the inter- arrival time probability distribution function is $1-(1-\omega)e^{-\lambda t}$. Thus, the arrival *point*-processes can be seen as batch-Poisson, with batches arriving at each point having geometric size distribution. The probability that a batch is of size s is $(1-\omega)\omega^{s-1}$.

The following notations are introduced.

- c is the number of servers.
- $I(t)$ denotes the number of busy servers at time t. Note that $I(t)$ varies within interval $[0, c]$.
- $J(t)$, which takes a value from 0 to ∞, represents the number of requests in the orbit at time t.

Service times are exponentially distributed with parameter μ. Clients which wait in the orbit retrial with rate ν (i.e.: the inter-repetition times are exponentially distributed with parameter ν). As a consequence, the system is modeled by Continuous Time Markov Chain (CTMC) $Y = \{I(t), J(t)\}$ with state space $\{0, 1, \ldots, c\} \times \{0, 1, \ldots\}$. We denote the steady state probabilities by $\pi_{i,j} = \lim\limits_{t\to\infty} Prob(I(t) = i, J(t) = j)$, and introduce $\mathbf{v}_j = (\pi_{0,j}, \ldots, \pi_{c,j})$.

The evolution of Y is driven by the following transitions.

(a) $A_j(i,k)$ denotes a transition rate from state (i,j) to state (k,j) $(0 \leqslant i, k \leqslant c; j = 0, 1, \ldots)$, which is caused by either the departure or the arrival of customers. Matrix A_j is defined as the matrix with elements $A_j(i,k)$.

$$A_j = A = \begin{bmatrix} 0 & \lambda(1-\omega) & \lambda(1-\omega)\omega & \ldots & & & \lambda(1-\omega)\omega^{c-1} \\ \mu & 0 & \lambda(1-\omega) & \ldots & & & \lambda(1-\omega)\omega^{c-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \ldots & (c-1)\mu & 0 & \lambda(1-\omega) \\ 0 & 0 & & \ldots & 0 & c\mu & 0 \end{bmatrix}$$

$\forall j \geqslant 0.$

(b) $B_{j,s}(i,k)$ represents s-steps upward transition from state (i,j) to state $(k,j+s)$ $(0 \leqslant i,k \leqslant c; s \geqslant 1; j=0,1,\ldots)$, which is due to the arrival of customers. In the similar way, matrix $B_{j,s}$ (B_s) with elements $B_{j,s}(i,k)$ is defined as

$$B_{j,s} = B_s = \begin{bmatrix} 0\,0\,0 \ldots 0\,0\,\lambda(1-\omega)\omega^{s+c-1} \\ 0\,0\,0 \ldots 0\,0\,\lambda(1-\omega)\omega^{s+c-2} \\ \vdots\,\vdots\,\vdots\,\vdots \quad \vdots\,\vdots \qquad \vdots \\ 0\,0 \quad \ldots 0\,0 \quad \lambda(1-\omega)\omega^{s} \\ 0\,0 \quad \ldots 0\,0 \;\; \lambda(1-\omega)\omega^{s-1} \end{bmatrix} \quad \forall j \geqslant 0; s \geqslant 1.$$

(c) $C_j(i,k)$ is the transition rate from state (i,j) to state $(k,j-1)$ $(0 \leqslant i,k \leqslant c; j=0,1,\ldots)$, which is due to the successful retry from the orbit. Matrix C_j $(\forall j \geqslant 1)$ with elements $C_j(i,k)$ is written as

$$C_j = C = \begin{bmatrix} 0\,\nu\,0 \ldots 0\,0\,0 \\ 0\,0\,\nu \ldots 0\,0\,0 \\ \vdots\,\vdots\,\vdots\,\vdots \quad \vdots\,\vdots\,\vdots \\ 0\,0 \quad \ldots 0\,0\,\nu \\ 0\,0 \quad \ldots 0\,0\,0 \end{bmatrix} \quad \forall j \geqslant 1.$$

D^A and D^C denotes diagonal matrices whose diagonal elements are the sum of the elements in the row of A and C. The following matrices are also introduced,

$$A^* = A - D^A,$$

$$\Lambda = Diag[\lambda\omega^c, \ldots, \lambda\omega, \lambda].$$

3. A Computational Procedure

For $j \geqslant 1$, the balance equations are written as follows

$$\sum_{s=1}^{j} \mathbf{v}_{j-s} B_s + \mathbf{v}_j \left[A^* - \Lambda - D^C\right] + \mathbf{v}_{j+1} C = 0.$$

For $j \geqslant 2$, we have

$$\sum_{s=1}^{j-1} \mathbf{v}_{j-1-s} B_s + \mathbf{v}_{j-1} \left[A^* - \Lambda - D^C\right] + \mathbf{v}_j C = 0,$$

therefore,

$$\mathbf{v}_{j-1} B_1 + \mathbf{v}_j \left[A^* - \Lambda - D^C\right] + \mathbf{v}_{j+1} C - \mathbf{v}_{j-1} \left[A^* - \Lambda - D^C\right] \omega - \mathbf{v}_j C \omega = 0,$$

$$\mathbf{v}_{j-1}(B_1 - \left[A^* - \Lambda - D^C\right] \omega) + \mathbf{v}_j(\left[A^* - \Lambda - D^C\right] - C\omega) + \mathbf{v}_{j+1} C = 0.$$

So, we arrive at the Quasi-Birth-and-Death (QBD) form as follows

$$\mathbf{v}_{j-1}Q_0 + \mathbf{v}_j Q_1 + \mathbf{v}_{j+1}Q_2 = 0 \ \ (j \geqslant 2), \tag{3.1}$$

where $Q_0 = (B_1 - \left[A^* - \Lambda - D^C\right]\omega)$, $Q_1 = (\left[A^* - \Lambda - D^C\right] - C\omega)$, $Q_2 = C$. Note that $Q(x) = Q_0 + Q_1 x + Q_2 x^2$ is defined as the characteristic matrix polynomial associated with equations (3.1). Due to the QBD form, the steady state probabilities can be obtained with the existing methods like the matrix-geometric and its variants [6, 11, 15], and the spectral expansion [13]. However, the existing methods have the numerical problem (no results due to a very long-running time of computer programs implementing these methods) when c is large (the problem starts when c reaches a value of several hundreds). Therefore, in what follows we present a fast computational procedure to find the steady state probabilities.

We have

$$Q(x) =$$

$$\begin{bmatrix} q_{11}(x) & (\omega - x)(\lambda(-1+\omega) - \nu x) & \cdots & & \lambda(-1+\omega)\omega^{c-2}(\omega - x) \\ \mu(x-\omega) & q_{2,2}(x) & \cdots & & \\ 0 & 2\mu(x-\omega) & (\omega - x)(\lambda(-1+\omega) - \nu x) & & \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & & (c-1)\mu(x-\omega) & q_{c,c}(x) & x(\lambda - \lambda\omega + \nu(-\omega + x)) \\ 0 & & 0 & c\mu(x-\omega) & q_{c+1,c+1}(x) \end{bmatrix}$$

where

$$\begin{aligned} q_{1,1}(x) &= (\lambda + \nu)(\omega - x), \\ q_{i,i}(x) &= (\lambda + i\mu + \nu)(\omega - x) \quad (i = 2, \ldots, c), \\ q_{c+1,c+1}(x) &= \lambda + c\mu(\omega - x) - \lambda x. \end{aligned}$$

The steady state probabilities are closely related to the eigenvalue-eigenvector pairs $(x, \boldsymbol{\psi})$ of $Q(x)$, which satisfy $\boldsymbol{\psi} Q(x) = 0$ and $det[Q(x)] = 0$ (c.f. [13]). Thus, the straightforward way to obtain the steady state probabilities is to find the eigenvalues of $Q(x)$ (see [5] for the methodology to find the eigensystem of the characteristic matrix polynomial). However, there exists an efficient method.

It is easy to see that $Q(x)$ has c eigenvalues of value ω. The corresponding independent eigenvectors for c eigenvalues are $\boldsymbol{\psi}_1 = \{1, 0, \ldots, 0\}$, $\boldsymbol{\psi}_3 = \{0, 1, 0, \ldots, 0\}$, ..., $\boldsymbol{\psi}_c = \{0, 0, \ldots, 1, 0\}$. Note that if the system is ergodic, then the number of eigenvalues of $Q(x)$, which are inside the unit disk, is $c+1$. Therefore, $Q(x)$ should have another eigenvalue called x_0 inside the unit disk. Let ψ_0 the corresponding left-hand-side eigenvector of $Q(\lambda)$ for the eigenvalue x_0.

As a consequence, the steady state probabilities can be expressed as follows

$$\begin{aligned} \mathbf{v}_j &= b_0 \boldsymbol{\psi}_0 x_0^j + \omega^j \sum_{i=1}^{c} b_i \boldsymbol{\psi}_i \ (j \geqslant 1) \\ &= b_0 \boldsymbol{\psi}_0 x_0^j + \omega^j \mathbf{b}, \end{aligned} \tag{3.2}$$

where b_i are the coefficients to be determined and $\mathbf{b} = \sum_{i=1}^{c} b_i \psi_i = \{b_1, b_2, \ldots, b_c, 0\}$.

Since the probabilities are greater than or equal to zero, $0 < x_0 < 1$ holds. Furthermore, $x_0 \neq \omega$ should hold to ensure that (c, j) states are reachable. It is observed that the key step towards the steady state probabilities is to determine x_0 and the corresponding eigenvector $\boldsymbol{\psi}_0$.

Theorem 3.1. $0 < x_0 < 1$ *is the root of* $l_{c+1}(x)$*, the last diagonal element of* $L(x)$ *when we make the LU decomposition of* $Q(x) = L(x)U(x)$.

Proof. Since $Q(x_0)$ is a tridiagonal matrix and $q_{i,i}(x_0) \neq 0$, the component matrices of the LU decomposition of $Q(x_0)$ are written as

$$L(x_0) = \begin{bmatrix} l_1(x_0) & 0 & 0 \dots 0 & 0 & 0 \\ \mu x_0 & l_2(x_0) & 0 \dots 0 & 0 & 0 \\ \vdots & \vdots & \vdots\ \vdots\ \vdots & \vdots & \vdots \\ 0 & 0 & \dots (c-1)\mu x_0 & l_c(x_0) & 0 \\ 0 & 0 & \dots 0 & c\mu x_0 & l_{c+1}(x_0) \end{bmatrix},$$

$$U(x_0) = \begin{bmatrix} 1 & u_{1,2} & \dots & u_{1,c-2} & u_{1,c} & u_{1,c+1} \\ 0 & 1 & u_{2,3} \dots & & u_{2,c} & u_{2,c+1} \\ \vdots & \vdots & \vdots\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \dots 0 & 1 & u_{c,c+1} \\ 0 & 0 & & \dots 0 & 0 & 1 \end{bmatrix},$$

where

$$\begin{aligned} l_1(x_0) &= q_{1,1}(x_0) = (\lambda + \nu)(\omega - x), \\ u_{1,i} &= q_{1,i}(x_0)/l_1(x_0) \quad (i = 2, \dots, c+1), \\ u_{j,i} &= (q_{j,i}(x_0) - q_{j,j-1}(x_0)u_{j-1,i})/l_j(x_0); (i = 2, \dots, c+1;\ j = 2, \dots, i-1), \\ l_i(x_0) &= q_{i,i}(x_0) - q_{i,i-1}u_{i-1,i} \quad (i = 2, \dots, c+1). \end{aligned}$$

Therefore, the determinant of $Q(x_0)$ is expressed as

$$Det[Q(x_0)] = Det[L(x_0)]Det[U(x_0)] = \prod_{i=1}^{c+1} l_i(x_0) \tag{3.3}$$

As the consequence of equation (3.3), we have $l_i(x_0) \neq 0$ $(1 < i \leqslant c)$. Hence, $Det[Q(x_0)] = 0$ follows $l_{c+1}(x_0) = 0$. □

It is also easy to prove that $l_{c+1}(0)$ is positive and $l_{c+1}(1)$ is negative. Therefore, a bisection algorithm in Figure 1 can be proposed to determine x_0 and $\boldsymbol{\psi}_0 = \{\psi_{0,1}, \psi_{0,2}, \dots, \psi_{0,c+1}\}$.

In what follows, we present a method to determine $\mathbf{b}$ and b_0. First, we prove that $\mathbf{b} = 0$ holds. We have $\boldsymbol{\psi}_0 Q(x_0) = 0$ because $(x_0, \boldsymbol{\psi}_0)$ is a eigenvalue/vector pair of $Q(x)$. This means,

$$\boldsymbol{\psi}_0(B_1 - [A^* - \Lambda - D^C]\omega + x_0([A^* - \Lambda - D^C] - C\omega) + x_0^2 C) = 0.$$

Algorithm 1 Bisection algorithm to determine x_0 and the calculation of ψ_0

```
Initialize the required accuracy $\epsilon$
$x_{0,u} = 1.0$, $x_{0,d} = 0$
repeat
  $x_0 = \frac{x_{0,u}+x_{0,d}}{2}$
  calculate $l_{c+1}(x_0)$ based on equation (3.3)
  if $l_{c+1}(x_0) > 0$ then
    $x_{0,d} = x_0$
  else
    $x_{0,u} = x_0$
  end if
until $|l_{c+1}(x_0)| < \epsilon$
$\psi_{0,1} = 1$
for $i = 1$ to $c$ do
  $\psi_{0,i+1} = \frac{\sum_{j=1}^{i} \psi_{0,i} q_{j,i}(x_0)}{i\mu(\omega - x_0)}$
end for
return $x_0$, $\psi_0$
```

After a simple algebra, we obtain

$$\boldsymbol{\psi}_0 B_1 + (x_0 - \omega)\boldsymbol{\psi}_0([A^* - \Lambda - D^C] + Cx_0) = 0. \tag{3.4}$$

$\boldsymbol{\psi}_0 B_1$ is a row vector with the first c zero-elements because B_1 is the matrix with the last nonzero-column. Therefore, due to (3.4), vector $\boldsymbol{\psi}_0([A^* - \Lambda - D^C] + Cx_0)$ should have the first c elements equal to zero.

We can write the balance equation for level 0 as

$$\mathbf{v}_0 \left[A^* - \Lambda\right] + \mathbf{v}_1 C = 0,$$

which follows

$$\begin{aligned}\mathbf{v}_0 &= \mathbf{v}_1 C[\Lambda - A^*]^{-1} \\ &= (b_0 \boldsymbol{\psi}_0 x_0 + \omega \mathbf{b}) C[\Lambda - A^*]^{-1}.\end{aligned} \tag{3.5}$$

Substituting (3.5) into the balance equation for level $J = 1$,

$$\mathbf{v}_0 B_1 + \mathbf{v}_1 \left[A^* - \Lambda - D^C\right] + \mathbf{v}_2 C = 0,$$

we obtain

$$\mathbf{v}_1 (C \left[\Lambda - A^*\right]^{-1} B_1 + \left[A^* - \Lambda - D^C\right]) + \mathbf{v}_2 C = 0.$$

Using (3.2), we get the following expression for $\mathbf{b}$ after some algebraic steps

$$\begin{aligned}(b_0 \boldsymbol{\psi}_0 x_0 + \omega \mathbf{b})(C[\Lambda - A^*]^{-1} B_1 + \left[A^* - \Lambda - D^C\right]) + (b_0 \boldsymbol{\psi}_0 x_0^2 + \omega^2 \mathbf{b}) C = 0, \\ b_0 \boldsymbol{\psi}_0 x_0 (C \left[\Lambda - A^*\right]^{-1} B_1 + \left[A^* - \Lambda - D^C\right] + x_0 C) +\end{aligned}$$

$$\begin{aligned}
\omega\mathbf{b}(C\,[\Lambda - A^*]^{-1}\,B_1 + \left[A^* - \Lambda - D^C\right] + \omega C) = 0,\\
-b_0\boldsymbol{\psi}_0 x_0(C[\Lambda - A^*]^{-1}B_1 + \left[A^* - \Lambda - D^C\right] + x_0 C) =\\
\omega\mathbf{b}(C[\Lambda - A^*]^{-1}B_1 + \left[A^* - \Lambda - D^C\right] + \omega C),\\
\mathbf{b} = -(b_0/\omega)\boldsymbol{\psi}_0 x_0(C[\Lambda - A^*]^{-1}B_1 + \left[A^* - \Lambda - D^C\right] + x_0 C)\\
(C(\Lambda - A^*)^{-1}B_1 + (A^* - \Lambda - D^C) + \omega C)^{-1}.
\end{aligned}$$

It is observed that $\boldsymbol{\psi}_0 x_0 C[\Lambda - A^*]^{-1}B_1$ is a row vector with the first c elements equal to zero because B_1 is the matrix with the last nonzero-column and recall that vector $\boldsymbol{\psi}_0([A^* - \Lambda - D^C] + Cx_0)$ has the first c elements equal to zero. As consequence $\mathbf{b}$ is the vector with the first c elements equal to zero, which means $\mathbf{b}$ is a zero-vector.

To determine coefficient b_0, we use the normalisation equation

$$1 = \sum_{i=0}^{c}\sum_{j=0}^{\infty}\pi_{i,j} = \mathbf{v}_0\mathbf{e} + \frac{b_0x_0}{1-x_0}\boldsymbol{\psi}_0\mathbf{e} = b_0x_0\boldsymbol{\psi}_0 C[\Lambda - A^*]^{-1}\mathbf{e} + \frac{b_0x_0}{1-x_0}\boldsymbol{\psi}_0\mathbf{e}.$$

4. Numerical Example

The proposed procedure is implemented in `Mathematica` (`http://www.wolfram.com`). We compare our algorithm and the solution of equation $det[Q(x)] = 0$ (i.e.: the direct way to determine the eigenvalues of the characteristic polynomial) with the following parameter values $\nu = 20$, $\omega = 0.26$, $\lambda = 2.3$ and $\mu = 1.0$. It is observed that our algorithm gives a correct result for root x_0 for all cases, while the direct solution of equation $det[Q(x)] = 0$ in Mathematica using a built-in function is not always correct.

- The built-in function of Mathematica finds that $det[Q(x)]$ has roots $x \rightarrow 0.26$, $x \rightarrow 0.26$, $x \rightarrow 0.26$, $x \rightarrow 0.26$, $x \rightarrow 0.859258$, $x \rightarrow 1$ and $x \rightarrow 10.3527$ when $c = 4$ holds. Our algorithm finds x_0 equal to 0.859258.

- With the built-in function of Mathematica $det[Q(x)]$ has roots $x \rightarrow 0.26 - 1.63875 \cdot 10^{-7}i$, $x \rightarrow 0.26 + 1.63875 \cdot 10^{-7}i$, $x \rightarrow 0.26$, $x \rightarrow 0.26$, $x \rightarrow 0.26$, $x \rightarrow 0.736433$, $x \rightarrow 1$, $x \rightarrow 5.93992$ and $x \rightarrow 55.9869$ when $c = 5$ holds. Note that $Q(x)$ does not have a complex eigenvalue in this case. Our algorithm results in $x_0 = 0.736433$.

The numerical results confirm a claim that we have developed a numerically stable algorithm for the solution the CPP/M/c retrial queue.

References

[1] Almási, B., Roszik, J., Sztrik, J., Homogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Mathematical and Computer Modelling*, (42):673–682, 2005.

[2] Artalejo, J.R., Economou, A., Lopez-Herrero, M.J., Algorithmic approximations for the busy period distribution of the M/M/c retrial queue, *European Journal of Operational Research*, 176:1687–1702, 2007.

[3] Artalejo, J.R., Pozo, M., Numerical Calculation of the Stationary Distribution of the Main Multiserver Retrial Queue, *Annals of Operations Research*, 1-4:41–56, 2002.

[4] Artalejo, J.R., Gómez-Corral, A., Retrial Queueing Systems, *Springer*, 2008.

[5] Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., H. van der Vorst, editors, Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, *SIAM*, Philadelphia.

[6] Bini, D., Meini, B., On the solution of a nonlinear matrix equation arising in queueing problems, *SIAM Journal on Matrix Analysis and Applications*, 17(4):906–926, 1996.

[7] Do, T.V., Chakka, R., Harrison, P.G., An integrated analytical model for computation and comparison of the throughputs of the UMTS/HSDPA user equipment categories, In MSWiM '07: *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 45–51, New York, NY, USA, 2007. ACM.

[8] Falin, G.I., Templeton, J.G.C., Retrial Queues, *Chapman & Hall*, London, 1997.

[9] Kouvatsos, D., Entropy maximisation and queueing network models, *Annals of Operations Research*, 48:63–126, 1994.

[10] Kumar, B., Raja, J., On multiserver feedback retrial queues with balking and control retrial rate, *Annals of Operations Research*, 141:211–232(22), January 2006.

[11] Latouch, G., Ramaswami, V., A logarithmic reduction algorithm for quasi-birth-death processes, *Applied Probability*, pages 650–674, 1993.

[12] Lopez-Herrero, M., A maximum entropy approach for the busy period of the M/G/1 retrial queue, *Annals of Operations Research*, 141:271–281(11), January 2006.

[13] Mitrani, I., Chakka, R., Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method, *Performance Evaluation*, 23:241–260, 1995.

[14] Mushko, V., Jacob, M., Ramakrishnan, K., Krishnamoorthy, A., Dudin, A., Multiserver queue with addressed retrials, *Annals of Operations Research*, 141:283–301(19), January 2006.

[15] Naoumov, V., Krieger, U., Wagner, D., Analysis of a Multi-server Delay-loss System with a General Markovian Arrival Process. In S.R. Chakravarthy and A.S. Alfa, editors, *Matrixanalytical methods in Stochastic models*, pages 43–66. Marcel Dekker, 1997.

[16] Roszik, J., Sztrik, J., Performance analysis of finite-source retrial queues with non-reliable heterogeneous servers, *Journal of Mathematical Sciences*, (146):6033–6038, 2007.

[17] Sztrik, J., Tool supported performance modelling of finite-source retrial queues with breakdowns, *Publicationes Mathematicae*, (66):197–211, 2005.

Tien Van Do
Department of Telecommunications
University of Technology and Economics
Budapest
Magyar tudósok krt. 2.
H-1117, Hungary
e-mail: `do@hit.bme.hu`

Annales Mathematicae et Informaticae
36 *(2009) pp. 71–75*
http://ami.ektf.hu

On the *k*-reversibility of finite automata

János Falucskai

College of Nyíregyháza, Institute of Mathematics and Informatics

Submitted 5 October 2009; Accepted 6 December 2009

Abstract

It is a famous result of Angluin (1982 [1]) that there exists a time polynomial and space linear algorithm to identify the canonical automata of k-reversible languages by using characteristic sample sets. This result has several applications. In this paper we characterise the class of all automata for which her method is not applicable. In particular, the aim of this paper is to characterise the family of finite automata which are not k-reversible for any non-negative integer k.

Keywords: finite automata, k-reversible automata

MSC: 68Q45, 68T50

1. Introduction

Without any doubt, there is no formal model that can capture all aspects of human learning. Nevertheless, the overall aim of researchers working in algorithmic learning theory has been to gain a better understanding of what learning really is. Several models are on the basis of the so-called learning autamata. Learning automata has a wide field of applications ranging over robotics and control systems, pattern recognition, computational linguistics, computational biology, data compression, data mining, etc. (see [5], for an excellent survey). Recently, learning techniques have also become popular in the area of automatic verification. They have been used [8] for minimizing (partially) specified systems and for model checking black-box systems, proved helpful in compositional model checking and in regular model checking. The general goal of learning algorithms employed in verification is to identify a machine, usually of minimal size, that conforms with an a priori fixed set of strings or a given machine. Nearly all algorithms learn deterministic finite-state automata (DFA) or deterministic finite-state machines (Mealy-/Moore machines), as the class of DFA has preferable properties in the setting of learning. For every regular language, there is a unique minimal DFA accepting it [6], which can be characterized by Nerode's right congruence [10, 9]. This characterization is at the base of most learning algorithms [5].

It is a famous result of Angluin [1] that there exists a time polynomial and space linear algorithm to identify the canonical automata of k-reversible languages by using characteristic sample sets. This result has various applications. (For example, the song learning of birds has similarity to the grammatical inference from positive samples [13] which works as Angluin's algorithm. Certain linguistic subsystems may also well be learnable by inductive inference method [12]. Her method is applicable in the natural language processing, too [4]).

The aim of this paper is to show the limitations of her method. In particular, we characterise the class of all automata which are not k-reversible for any non-negative integer k. The author did not find any paper studying or characterising the class of automata having this property. In other words, it has a high likehood that there are no related works regarding our results.

2. Preliminaries

We start with some standard concepts and notations. All concepts not defined here can be found in [3, 6].

By an *automaton* we mean a finite Rabin-Scott automaton, i.e. a deterministic finite initial automaton without outputs supplied by a set of final states which is a subset of the state set. In more details, an automaton is an algebraic structure $\mathcal{A} = (A, a_0, A_F, \Sigma, \delta)$ consisting of the nonempty and finite *state set* A, the nonempty and finite *input set* Σ, a *transition function* $\delta : A \times \Sigma \to A$, the initial state $a_0 \in A$ and the (not necessarily nonempty) set $A_F \subseteq A$ of final states.

It is understood that δ is extended to $\delta^* : A \times X^* \to A$ with $\delta^*(a, \lambda) = a,\ \delta^*(a, xq) = \delta(a, x)\delta^*(\delta(a, x), q), a \in A, x \in \Sigma, q \in \Sigma^*$. In other words, $\delta^*(a, \lambda) = a$ and for every nonempty input word $x_1x_2 \cdots x_s \in \Sigma^+$ (where $x_1, x_2, \ldots, x_s \in \Sigma$) there are $a_1, \ldots, a_s \in A$ with $\delta(a, x_1) = a_1, \delta(a_1, x_2) = a_2, \ldots, \delta(a_{s-1}, x_s) = a_s$ such that $\delta^*(a, x_1 \cdots x_s) = a_1 \cdots a_s$.

Moreover, for every $a \in A, w \in \Sigma^*$, denote by $a \cdot w$ the last letter of $\delta^*(a, w)$. The concept of *acceptor* is a natural generalization of the concept of automaton. By an *acceptor* we mean a system $\mathbf{A} = (A, I, F, \Sigma, \delta)$ such that A is a finite (not necessarily nonempty) set, the set of *states, $I \subseteq A$* is the set of *initial states, $F \subseteq A$* is the set of *final* or *accepting states* and $\delta : A \times \Sigma \to 2^A$ is the *transition function.* $\mathbf{A}$ is called *deterministic* if $|I| \leqslant 1$ and for every $a \in A, x \in X$, $|\delta(a, x)| \leqslant 1$. Thus an automaton can be considered as a special deterministic acceptor. The *reverse* of an acceptor $\mathbf{A} = (A, I, F, \Sigma, \delta)$ is the acceptor $\mathbf{A}^r = (A, F, I, \Sigma, \delta^r)$ having $\delta^r(a, x) = \{b \in A \mid a \in \delta(b, x)\}$ for all $a \in A, x \in \Sigma$. An acceptor $\mathbf{A}$ is called *zero reversible* if both of $\mathbf{A}$ and $\mathbf{A}^r$ are deterministic. $\mathbf{A}$ is k-reversible for a positive integer k if $\mathbf{A}$ is deterministic, moreover, for any pair $a_1, a_2 \in A, a_1 \neq a_2$, if $a_1, a_2 \in F$ or $a_1, a_2 \in \delta^r(a, x)$ for some $a \in A$ and $x \in \Sigma$, then for every $w \in \Sigma^*, |w| = k$, at least one of $\delta^r(a_1, w), \delta^r(a_2, w)$ should be $\emptyset$. It is said that the acceptor $\mathbf{A}$ *accepts the empty word* if there exists an $a \in I$ with $a \in F$. Furthermore, we say that $\mathbf{A}$ accepts a nonempty word $x_1 \cdots x_s \in \Sigma^+$ $(x_1, \ldots, x_s \in \Sigma)$ if there are $a_1, \ldots, a_{s+1} \in A$ with $a_1 \in I, a_{s+1} \in F$, and $a_{i+1} \in \delta(a_i, x_i), i = 1, \ldots, s$. The language $L_{\mathbf{A}} \subseteq \Sigma^*$ consisting of all words in Σ^* accepted by $\mathbf{A}$ is called the language *accepted by* $\boldsymbol{A}$. A language $L \subseteq \Sigma^*$ is said to be k-reversible for some nonnegative integer k, if there exists a k-reversible acceptor $\mathbf{A}$ with $L = L_{\mathbf{A}}$. A deterministic acceptor $\mathbf{A} = (A, I, F, \Sigma, \delta_{\mathbf{A}})$ with $|I| = 1$ and $\forall a \in A, x \in \Sigma : |\delta_{\mathbf{A}}(a, x)| = 1$ can be considered as the automaton $\mathcal{A} = (A, a_0, A_F, \Sigma, \delta_{\mathcal{A}})$ with $\{a_0\} = I, A_F = F$, $\forall a \in A, x \in \Sigma : \{\delta_{\mathcal{A}}(a, x)\} = \delta_{\mathbf{A}}(a, x)$ and vice versa. Thus we can extend the concept of k-reversibility to automata in a natural way.

3. Results

The following statement can be derived directly from the definition of k-reversibility of automata (with the notations $a = a_1, b = a_2, u = w, c = \delta^r(a_1, w), d = \delta^r(a_2, w)$).

Lemma 3.1. *Given a nonnegative integer k, the automaton $\mathcal{A} = (A, a_0, A_F, \Sigma, \delta)$ is k-reversible if and only if for every distinct $a, b \in A$, there do not exist $c, d \in A$, $u \in \Sigma^*$ with $|u| = k$, having $c \cdot u = a, d \cdot u = b$ whenever $a, b \in A_F$ or $\delta(a, x) = \delta(b, x)$ for some $x \in \Sigma$.*

Next, we prove the following Theorem:

Theorem 3.2. *Let $\mathcal{A} = (A, a_0, A_F, \Sigma, \delta)$ be an arbitrary automaton. There does not exist a nonnegative integer k for which $\mathcal{A}$ is k-reversible if and only if there are distinct states $a, b \in A$, a nonempty input word $u \in \Sigma^+$, an input word $v \in \Sigma^*$, such that $a \cdot u = a, b \cdot u = b, a \cdot v \neq b \cdot v$, and either $a \cdot v, b \cdot v \in A_F$ or $a \cdot vx = b \cdot vx$ for some $x \in \Sigma$.*

Proof. First, we suppose that there are distinct states $a, b \in A$, a nonempty input word $u \in \Sigma^+$, an input word $v \in \Sigma^*$ such that $a \cdot u = a, b \cdot u = b, a \cdot v \neq b \cdot v$, and either $a \cdot v, b \cdot v \in A_F$ or $a \cdot vx = b \cdot vx$ for some $x \in \Sigma$. Assume that, contrary of our statement, $\mathcal{A}$ is k-reversible for some nonnegative integer k. By $a \cdot u = a, b \cdot u = b, a \neq b, u \neq \lambda$ and Lemma 3.1, this is impossible if $a, b \in A_F$. Therefore, at least one of a and b should be a non-final state. Thus, by our conditions, there is an $x \in \Sigma$ with $a \cdot vx = b \cdot vx$. On the other hand, by $a \cdot vx = b \cdot vx$, it is clear that $k > 0$. Now, let $k > 0$ and consider the minimal nonnegative integer ℓ with $|u^\ell v| \geqslant k$. First, we prove that for every prefix w of $u^\ell v$, $a \cdot w \neq b \cdot w$. If $u = wz$ for some $z \in \Sigma^*$, then $a \cdot w = b \cdot w$ implies $\delta(a \cdot wz) = \delta(b \cdot wz)$ which leads to $(a =)$ $a \cdot u = b \cdot u$ $(= b)$, which is a contradiction. Now, let i, j be nonnegative integers such that $w = u^{i+j} z$ and $u^j z$ is a prefix of v. First, $a \cdot u^i = a \neq b = b \cdot u^i$ holds, because of $a \cdot u = a, b \cdot u = b$ with $a \neq b$. On the other hand, $v = u^j zr$ for some $r \in \Sigma^*$, because $u^j z$ is a prefix of v. Therefore, using $a \cdot u^i = a, b \cdot u^i = b$, if $a \cdot u^{i+j} z = b \cdot u^{i+j} z$, then $a \cdot u^j z = b \cdot u^j z$ leading to

$a \cdot u^j zr = b \cdot u^j zr$ with $u^j zr = v$, which is a contradiction. Consider $w, z \in \Sigma^*$ with $u^\ell v = wz$ and $|z| = k$. We have already proved $a \cdot w \neq b \cdot w$. On the other hand, by our assumptions, $a \cdot wz \neq b \cdot wz$ and $\delta(a \cdot wzx) = \delta(b \cdot wzx)$. By Lemma 3.1, considering $a \cdot w, b \cdot w, a \cdot wz, b \cdot wz, z, x$ as c, d, a, b, u, x, we obtain that $\mathcal{A}$ is not k-reversible. Now, we assume that for every nonnegative integer k, the automaton $\mathcal{A}$ is not k-reversible. This means that $\mathcal{A}$ is not 0-reversible. Moreover, by Lemma 3.1, for every positive integer k, there are distinct $a, b \in A$, such that there exist $c, d \in A$, $u \in \Sigma^*$ with $|u| = k$, $c \cdot u = a, d \cdot u = b$, where we have either $a, b \in A_F$ or $\delta(a, x) \neq \delta(b, x)$ for some $x \in \Sigma$. Without any restriction we may assume that $k \geqslant |A|^2$. Obviously, by $a \neq b$, for every prefix w of u, $c \cdot w \neq d \cdot w$. But then, by $(k =)|u| > \frac{|A|(|A|-1)}{2}, u = x_1 \cdots x_k$ with $x_1, \ldots, x_k \in \Sigma$, there exists a repetition in the sequence $(c, d), (\delta(c, x_1), \delta(d, x_1)), (c \cdot x_1 x_2, d \cdot x_1 x_2), \ldots, (c \cdot x_1 \cdots x_k, d \cdot x_1 \cdots x_k)$ having $c \neq d$ and $c \cdot x_1 \cdots x_i \neq d \cdot x_1 \cdots x_i, i = 1, \ldots, k$. Thus, there are $p, r \in \Sigma^*, q \in \Sigma^+$ with $u = pqr$ and $c \cdot p = c \cdot pq, d \cdot p = d \cdot pq, c \cdot pq \neq d \cdot pq, c \cdot pqr \neq d \cdot pqr$ and either $c \cdot pqr, d \cdot pqr \in A_F$ or $c \cdot pqrx = d \cdot pqrx$ for some $x \in \Sigma$. By Lemma 3.1, this shows that $\mathcal{A}$ is not k-reversible. □

4. Conclusion

It is well-known that, by using characteristic sample sets, the canonical automata of k-reversible languages can be identified applying a time polynomial and space linear algorithm (this is a famous result of Angluin). In this paper the limitations of her method are shown. In other words, the characterisation is given for automata which are not k-reversible for any non-negative integer k. It is an interesting fact that this property was not investigated so far in the literature. Further work is to characterise classes of automata and their languages for which other learning algorithms can not be applied [2, 14, 7, 11].

References

[1] Angluin, D., Inference of reversible languages, *J. Assoc. Comput. Mach.*, 29 (1982) 741–765.

[2] Biermann, A. W., Feldman, J. A., On the synthesis of finite-state machines from samples of their behaviour, *IEEE Transactions on Computers*, 21 (1972) 592–597.

[3] Dömösi, P., Nehaniv, C. L., Algebraic theory of automata networks. An introduction. *SIAM Monographs on Discrete Mathematics and Applications*, 11 (2005).

[4] García, P., Vázquez de Parga, M., López, D., On the efficient construction of quasi-reversible automata for reversible languages, *Inform. Process. Lett.*, 107 (2008) 13–17.

[5] Higuera, C., A bibliographical study of grammatical inference. *Pattern Recognition*, 38 (9) (2005) 1332–1348.

[6] Hopcroft, J.E., Motwani, R., Ullman, J. D., Introduction to Automata Theory (second edition), *Addison-Wesley Series in Computer Science*, (2001).

[7] Lang, K., Random DFA's can be approximately learned from sparse uniform examples, *COLT*, (2001).

[8] Leucker, M., Learning meets verification, *FMCO'07, LNCS*, 4709 (2007) 127–151.

[9] Myhill, J., Finite automata and the representation of events, *Tech. Rep. WADD TR-57-624, Wright Patterson Air Force Base, Ohio*, (1957).

[10] Nerode, A., Linear automaton transformations, *Proc. of the American Mathematical Society*, 9 (1958) 541–544.

[11] [Oncina, J., Garcia P., Inferring regular languages in polynomial update time, *Pattern Recognition and Image Analysis, volume 1 of Series in Machine Perception and Artificial Intelligence*, (1992).

[12] Pilato, S., Berwick, R., Reversible automata and induction of the english auxiliary system. *Proceedings of the ACL*, (1985) 70–75.

[13] Sasahara, K., Kakishita, Y., Nishino, T., Takahasi M., Okanoya, K., Constructing song syntax by automata induction *Lecture Notes in Artificial Intelligence*, 4201 (2006) 351–353.

[14] Trakhtenbrot, B.A., Barzdin, J.M., Finite automata: behaviour and synthesis, *North-Holland*, (1973).

János Falucskai
Nyíregyháza
Sóstói út 31/B
H-4400
Hungary
e-mail: `falu@nyf.hu`

Annales Mathematicae et Informaticae
36 *(2009) pp. 77–84*
http://ami.ektf.hu

Surface interpolation with local control by linear blending

Imre Juhász[a], Miklós Hoffmann[b]

[a]University of Miskolc, Hungary

[b]Károly Eszterházy College, Hungary

Submitted 25 May 2009; Accepted 10 December 2009

Abstract

The purpose of this paper is to introduce an interactive surface interpolation method by spline surfaces, which is a generalization of the method presented in [2]. The technique is based on linear blending and works for a large class of surfaces including bicubic Bézier, B-spline, NURBS surfaces and the recently developed trigonometric surfaces as well. The interpolating surface can be interactively modified by control points, meanwhile the interpolation property is preserved.

Keywords: interpolation, spline surface, linear blending

MSC: 65D17, 68U07

1. Introduction

Interpolation of an ordered set of points is one of the most widely used methods in curve and surface modeling practice, hence there is a vast number of papers and book chapters dealing with this topic (cf. the books [6, 15] and references therein). Designers generally prefer splines, where most of the methods work globally. Even in the case of B-spline or NURBS surfaces, which are standard description methods in geometric design and have local control properties, the process of finding control points of an interpolating surface is global and the resulted surface cannot be locally controlled (see e.g. [1, 10, 13, 7]). To overcome this problem some methods have been developed by means of which the shape of the interpolating curve or surface can be adjusted by numerical techniques (see e.g. [8, 9, 16]). Shape parameters and other numerical techniques, however, do not provide intuitive shape control methods such as control point repositioning in approximation. Moreover, in these global methods a large system of equations has to be solved at a relatively high computational cost. Especially, for large set of data points, local methods have the advantage of solving smaller systems, since the computation of each curve segment is based on only a subset of data. Unfortunately, these local methods typically attain only C^1 continuity [11].

In the last couple of years a new local method has been developed for some types of spline curves and surfaces that requires only local computation and yields C^2 continuous spline curves. This technique - which is based on linear blending - has been implemented for NURBS in [17], for B-spline curve in [18] and for trigonometric C-B-spline curve in [14]. In this method the shape of the interpolating curve can also be adjusted numerically by some shape parameters.

In [2] the authors generalized the linear blending interpolation method for a large class of curves. The present contribution is the further generalization of the linear blending curve concept for surfaces. Since designers generally prefer geometric entities instead of numerical values, we provide intuitive, control point based modification of the interpolating surface. We let the designer alter the shape of the surface similarly to the approximating surfaces, meanwhile the interpolation property is continuously preserved.

2. Linear blending surfaces

At first we describe a surface generating method which we will call linear blending. Consider the points $\mathbf{p}_{k,l}$, $(k = 0, \dots, n; l = 0, \dots, m)$ and the piecewisely defined surface

$$\mathbf{b}(u,v) = \sum_{k=-1}^{n+1} \sum_{l=-1}^{m+1} F_k(u) G_l(v)\, \mathbf{p}_{kl}, \quad u \in [u_0, u_n], v \in [v_0, v_m], \tag{2.1}$$

where $\mathbf{p}_{kl}$ are called control points (they form the control net consisting of quadrilateral "faces"), $F_k(u)$ and $G_l(v)$ are basis functions of some space (not necessarily of the same). The number of faces must be equal to the number of patches. Thus, in case of open surfaces one has to define artificial control points, e.g. by doubling the control points on the boundary of the control net. The only restriction is that each patch of the surface has to be defined by 16 (4×4) neighboring control points, that is patches of this surface can be written as

$$\mathbf{b}_{i,j}(u,v) = \sum_{k=i-1}^{i+2} \sum_{l=j-1}^{j+2} F_k(u) G_l(v)\, \mathbf{p}_{kl}, \quad u \in [u_i, u_{i+1}], v \in [v_j, v_{j+1}]$$
$$i = 0, \dots, n-1, j = 0, \dots, m-1$$

where the values u_i, v_j are called knots.

Now, consider the (i,j)th face of the control net determined by the control points $\mathbf{p}_{i,j}, \mathbf{p}_{i,j+1}, \mathbf{p}_{i+1,j}, \mathbf{p}_{i+1,j+1}$ and interpolate this quadrilateral with the double ruled surface that we obtain from the four corner points by bilinear combination with some functions $f(u)$ and $g\,(v)$ in the form

$$\mathbf{h}_{i,j}(u,v) = \begin{bmatrix}(1-f(u)) \; f(u)\end{bmatrix}\begin{bmatrix}\mathbf{p}_{i,j} & \mathbf{p}_{i,j+1}\\ \mathbf{p}_{i+1,j} & \mathbf{p}_{i+1,j+1}\end{bmatrix}\begin{bmatrix}(1-g(v))\\ g(v)\end{bmatrix},$$
$$u \in [u_i, u_{i+1}]\,, v \in [v_j, v_{j+1}]\,.$$

If the four points are coplanar this surface degenerates to a quadrilateral region in their plane, otherwise it is a patch of a hyperbolic paraboloid (saddle).

For the sake of interpolation, functions $f(u), g(v)$ have to fulfill conditions

$$f(u_i) = g(v_j) = 0, \quad f(u_{i+1}) = g(v_{j+1}) = 1. \tag{2.2}$$

Linearly blending the patches $\mathbf{b}_{i,j}\,(u,v)$ with the corresponding double ruled patches $\mathbf{h}_{i,j}(u,v)$ we obtain the linear blending surface consisting of the patches

$$\mathbf{c}_{i,j}(u,v,\alpha) = (1-\alpha)\mathbf{b}_{i,j}\,(u) + \alpha\mathbf{h}_{i,j}(u),$$

where α is a global shape parameter of the surface.

To achieve more flexibility in shape modification, the shape parameter α can also be the function of u and v. A natural way is to define them piecewisely by local shape parameters $\alpha^*_{i,j}$ associated to each point $\mathbf{p}_{i,j}$, and to use the same blending functions $f(u)$ and $g(v)$ like for the surface $\mathbf{h}_{i,j}(u,v)$, i.e.,

$$\alpha_{i,j}(u,v) = \begin{bmatrix}(1-f(u)) \; f(u)\end{bmatrix}\begin{bmatrix}\alpha^*_{i,j} & \alpha^*_{i,j+1}\\ \alpha^*_{i+1,j} & \alpha^*_{i+1,j+1}\end{bmatrix}\begin{bmatrix}(1-g(v))\\ g(v)\end{bmatrix}.$$

In this way each patch of the linear blending surface will have four local shape parameters (these parameters initially can be defined to be equal to 1 and will be modified in an interactive way)

$$\begin{gathered}\mathbf{c}_{i,j}(u,v,\alpha^*_{i,j},\alpha^*_{i,j+1},\alpha^*_{i+1,j},\alpha^*_{i+1,j+1}) =\\ (1-\alpha_{i,j}(u,v))\mathbf{b}_{i,j}\,(u,v) + \alpha_{i,j}(u,v)\mathbf{h}_{i,j}(u,v).\end{gathered} \tag{2.3}$$

In order to obtain $C^r (r > 0)$ continuity at joints of consecutive patches, functions $f(u), g(v)$ have to satisfy the conditions

$$\begin{array}{l} f^{(k)}(u_i) = f^{(k)}(u_{i+1}) = 0 \quad 1 \leqslant k \leqslant r, \quad j = 0,\ldots,n-1\\ g^{(k)}(v_j) = g^{(k)}(v_{j+1}) = 0 \quad 1 \leqslant k \leqslant r, \quad j = 0,\ldots,m-1 \end{array}. \tag{2.4}$$

Beside these conditions, the choice of these functions highly depends on the base surface $\mathbf{b}(u,v)$, more precisely the type of its basis functions $F_k\,(u)\,, G_l\,(v)$. The possible choice and generalizations of these functions can be found in [2].

3. Interpolation by linear blending surfaces

Now, we generalize the idea of curve interpolation by linear blending described in [17] for surfaces, and we modify the linear blending method defined above to interpolate a given grid of points. Surface patch $\mathbf{c}_{i,j}(u,v,\alpha^*_{i,j},\alpha^*_{i,j+1},\alpha^*_{i+1,j},\alpha^*_{i+1,j+1})$

is "between" the patches $\mathbf{b}_{i,j}(u,v)$ and $\mathbf{h}_{i,j}(u,v)$, that is the patch approximates the given points $\mathbf{p}_{i,j}, \mathbf{p}_{i,j+1}, \mathbf{p}_{i+1,j}, \mathbf{p}_{i+1,j+1}$. Note, that the method works in that cases as well when the double ruled surface is determined by any four points. We are going to specify such four corner points for the double ruled surface $\mathbf{h}_{i,j}(u,v)$ that the resulted blending patch $\mathbf{c}_{i,j}$ will interpolate the given points.

Let the points $\mathbf{p}_{i,j}$, associated parameter values (u_i, v_j) and shape parameters $\alpha^*_{i,j}$, $(i = 0, \ldots, n, j = 0, \ldots, m)$ be given. The problem is to find a linear blending patch that has the given shape parameters $\alpha^*_{i,j}$ and interpolates the given points $\mathbf{p}_{i,j}$ at the given parameter values (u_i, v_j).

Let us consider the approximating patch defined by the given points as control points

$$\mathbf{b}_{i,j}(u,v) = \sum_{k=i-1}^{i+2} \sum_{l=j-1}^{j+2} F_k(u) G_l(v) \mathbf{p}_{kl}, \quad u \in [u_i, u_{i+1}], v \in [v_j, v_{j+1}],$$

and the double ruled patch

$$\mathbf{h}_{i,j}(u,v) = \begin{bmatrix} (1-f(u)) & f(u) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{i,j} & \mathbf{v}_{i,j+1} \\ \mathbf{v}_{i+1,j} & \mathbf{v}_{i+1,j+1} \end{bmatrix} \begin{bmatrix} (1-g(v)) \\ g(v) \end{bmatrix}, \\ u \in [u_i, u_{i+1}], v \in [v_j, v_{j+1}]$$

where the points $\mathbf{v}_{i,j}$ are unknown. Using the interpolation assumptions

$$\begin{aligned} \mathbf{c}_{i,j}(u_i, v_j, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1}) &= \mathbf{p}_{i,j} \\ \mathbf{c}_{i,j}(u_i, v_{j+1}, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1}) &= \mathbf{p}_{i,j+1} \\ \mathbf{c}_{i,j}(u_{i+1}, v_j, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1}) &= \mathbf{p}_{i+1,j} \\ \mathbf{c}_{i,j}(u_{i+1}, v_{j+1}, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1}) &= \mathbf{p}_{i+1,j+1} \end{aligned} \tag{3.1}$$

we obtain

$$\begin{aligned} \mathbf{v}_{i,j} &= \mathbf{p}_{i,j} + \frac{1-\alpha^*_{i,j}}{\alpha^*_{i,j}} (\mathbf{p}_{i,j} - \mathbf{b}_{i,j}(u_i, v_j)) \\ \mathbf{v}_{i,j+1} &= \mathbf{p}_{i,j+1} + \frac{1-\alpha^*_{i,j+1}}{\alpha^*_{i,j+1}} (\mathbf{p}_{i,j+1} - \mathbf{b}_{i,j}(u_i, v_{j+1})) \\ \mathbf{v}_{i+1,j} &= \mathbf{p}_{i+1,j} + \frac{1-\alpha^*_{i+1,j}}{\alpha^*_{i+1,j}} (\mathbf{p}_{i+1,j} - \mathbf{b}_{i,j}(u_{i+1}, v_j)) \\ \mathbf{v}_{i+1,j+1} &= \mathbf{p}_{i+1,j+1} + \frac{1-\alpha^*_{i+1,j+1}}{\alpha^*_{i+1,j+1}} (\mathbf{p}_{i+1,j+1} - \mathbf{b}_{i,j}(u_{i+1}, v_{j+1})). \end{aligned} \tag{3.2}$$

By means of these points, the corresponding linear blending patches will interpolate the given points at the given parameter values (see Fig.1).

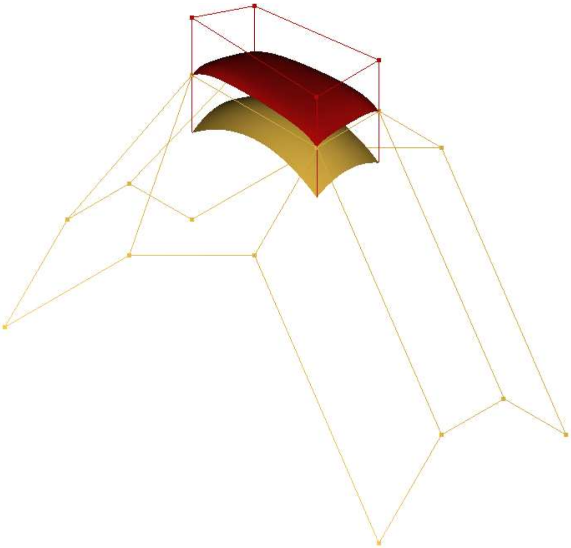


Figure 1: The original base surface (below) and the interpolating linear blending surface with the calculated four control points.

4. Interactive shape modification

Points $\mathbf{v}_{i,j}$ depend on three parameters: the corresponding parameter values u_i, v_j and the local shape parameter $\alpha^*_{i,j}$. Instead of manipulating these values numerically, calculating the points $\mathbf{v}_{i,j}$ and finally the interpolating surface patch $\mathbf{c}_{i,j}(u, v, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1})$, we intend to develop an interactive shape modification tool. In this tool points $\mathbf{v}_{i,j}$ shall be used analogously to the control points of an approximating surface, meanwhile the interpolating property of the surface is preserved. Although, these points are not "real" control points of the surface, the geometric effect of dragging these points is quite similar to the effect of control point repositioning.

When the position of the point $\mathbf{v}_{i,j}$ is modified, we have to recalculate the actual values of parameters u_i, v_j and $\alpha^*_{i,j}$ to preserve the interpolation. This problem leads us to the following questions: what happens to the surface (and especially to the point $\mathbf{v}_{i,j}$) if one of these parameters is changed? What are the possible positions of the point $\mathbf{v}_{i,j}$?

At first let us fix the parameters u_i, v_j and alter the shape parameter $\alpha^*_{i,j}$. It is obvious from Eq. (3.2) that preserving the interpolation property the point $\mathbf{v}_{i,j}$ will move along a straight line connecting the given point $\mathbf{p}_{i,j}$ and the point $\mathbf{b}_{i,j}(u_i, v_j)$ of the original surface patch.

Now, consider the case when the shape parameter $\alpha^*_{i,j}$ is fixed and the parameters u_i, v_j are altered. By Eqs. (3.1) the surface interpolates the point $\mathbf{p}_{i,j}$

at parameters (u_i, v_j), which may vary between u_{i-1}, u_{i+1} and v_{j-1}, v_{j+1}, respectively. These values, however also serve as knot values of the original base surface (2.1). Therefore, the alteration of these parameters changes the shape of the original surface patch $\mathbf{b}_{i,j}(u, v)$ as well. The geometric description of the effect of knot alteration is far from being trivial. For B-spline and NURBS surfaces it has been described in detail in [3], [4], [12] and [5]. Using the results of these studies we can conclude that the point $\mathbf{b}_{i,j}(u_i, v_j)$ of the base surface will move along a well-defined surface patch

$$\mathbf{e}(u_i, v_j) = \mathbf{b}_{i,j}(u_i, v_j) \quad u_i \in [u_{i-1}, u_{i+1}], v_j \in [v_{j-1}, v_{j+1}]. \tag{4.1}$$

E.g., in case of a B-spline surface of degree (k, l), the surface $\mathbf{e}(u_i, v_j)$ is a B-spline surface patch of degree $(k-1, l-1)$, defined by the same control points and knot values (except the knots u_i and v_j) as the original surface [3].

By means of Eqs. (3.2) it is easy to see that altering the parameters u_i, v_j the point $\mathbf{v}_{i,j}$ will move along a surface that can be obtained by a central similarity from surface (4.1), where the center of similitude is the given point $\mathbf{p}_{i,j}$ and the ratio is $\left(1 - \alpha^*_{i,j}\right) / \alpha^*_{i,j}$.

Summarizing the above results one can see that the permissible positions of $\mathbf{v}_{i,j}$ is a volume bounded by a cone-like surface the apex of which is the given point $\mathbf{p}_{i,j}$ and its base is composed of the four boundary curves of the envelope surface (4.1) (see Fig.2).

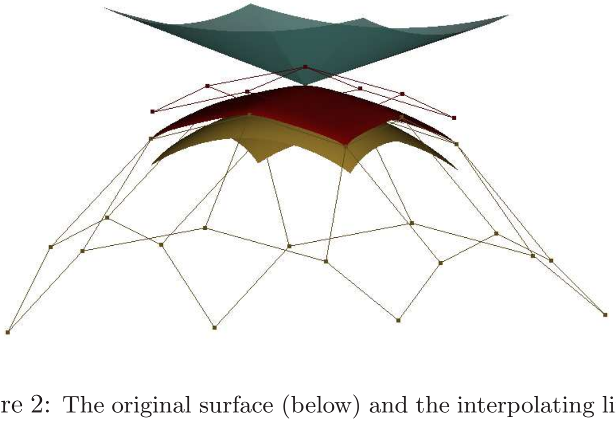


Figure 2: The original surface (below) and the interpolating linear blending surface. The permissible positions of the upmost control point is shown by a volume bounded by a cone-like surface.

For each actual position of $\mathbf{v}_{i,j}$ within this region one has to recalculate the parameters u_i, v_j and $\alpha^*_{i,j}$, and (by fixing the rest of the shape parameters) substitute them into $\mathbf{c}_{i,j}(u, v, \alpha^*_{i,j}, \alpha^*_{i,j+1}, \alpha^*_{i+1,j}, \alpha^*_{i+1,j+1})$ in order to obtain the interpolating surface.

5. Conclusions

An easy-to-compute interpolation method is presented in this paper, based on linear blending of a base surface and a computed control mesh. The resulted surface can interactively be modified by the points of this control mesh, meanwhile the interpolation property continuously holds. The method works for a large class of surfaces, including all the standard surface types (Bézier, B-spline, NURBS, C-B-spline, etc.) of computer aided geometric design.

Acknowledgements. The authors wish to thank the National Office of Research and Technology (Project CHN-37/2005) for their financial support of this research. The second author is also supported by János Bolyai Scholarship of Hungarian Academy of Sciences.

References

[1] Barsky, B.A., Greenberg, D.P., Determining a set of B-spline control vertices to generate an interpolating surface, *Computer Graphics and Image Processing*, 14 (1980), 203–226.

[2] Hoffmann, M., Juhász, I., On interpolation by spline curves with shape parameters, *Lecture Notes in Computer Science*, 4975 (2008), 205–214.

[3] Hoffmann M., Juhász I., Geometric aspects of knot modification of B-spline surfaces, *Journal for Geometry and Graphics*, 6 (2002), 141–149.

[4] Hoffmann, M., Juhász, I., On the family of B-spline surfaces obtained by knot modification, *Mathematical Communications*, 11 (2006), 9–16.

[5] Hoffmann, M., Juhász, I., Constrained shape control of bicubic B-spline surfaces by knots, in: Sarfraz, M., Banissi, E. (Eds.) Geometric Modeling and Imaging, London, IEEE CS Press, 41–47, 2006

[6] Hoschek, J., Lasser, D., Fundamentals of CAGD, AK Peters, Wellesley, MA, 1993.

[7] Juhász, I., Hoffmann, M., On parametrization of interpolating curves, *Journal of Computational and Applied Mathematics*, 216 (2008), 413–424.

[8] Kaklis, P.D., Sapidis, N.S., Convexity-preserving interpolatory parametric splines of non-uniform polynomial degree, *Computer Aided Geometric Design*, 12 (1995), 1–26.

[9] Kong, V.P., Ong, B.H., Shape preserving F3 curve interpolation, *Computer Aided Geometric Design*, 19 (2002), 239–256.

[10] Lavery, J.E., Univariate cubic Lp splines and shape-preserving multiscale interpolation by univariate cubic L1 splines, *Computer Aided Geometric Design*, 17 (2000), 319–336.

[11] Li, A., Convexity preserving interpolation, *Computer Aided Geometric Design*, 16 (1999), 127–147.

[12] Li, Y.-J., Wang, G.-Z., On knot modification of B-spline and NURBS surfaces. *J. of Computer-Aided Design and Computer Graphics*, 15 (2005), 986–989.

[13] Ma, W., Kruth, J.P., NURBS curve and surface fitting and interpolation, in: M. Daehlen, T. Lyche, L. Schumaker (Eds.), Mathematical Methods for Curves and Surfaces, Vanderbilt University Press, Nashville & London, 1995.

[14] Pan, Y.-J., Wang, G.-J., Convexity-preserving interpolation of trigonometric polynomial curves with shape parameter, *Journal of Zhejiang University Science*, 8 (2007), 1199–1209.

[15] Piegl, L., Tiller, W., The NURBS Book, Springer, Berlin. 1995

[16] Sapidis, N., Farin, G., Automatic fairing algorithm for B-spline curves, *Computer-Aided Design*, 22 (1990), 121–129.

[17] Tai, C.-L., Barsky, B.A., Loe, K.-F., An interpolation method with weights and relaxation parameters. In: Cohen, A., Rabut, C., Schumaker, L.L. (eds.): Curve and Surface Fitting: Saint-Malo 1999. Vanderbilt Univ. Press, Nashville, Tenessee (2000) 393–402.

[18] Tai, C.-L., Wang, G.-J., Interpolation with slackness and continuity control and convexity-preservation using singular blending, *J. Comp. Appl. Math.*, 172 (2004), 337–361.

Imre Juhász
Department of Descriptive Geometry, University of Miskolc,
H-3515 Miskolc, Hungary
e-mail: agtji@uni-miskolc.hu

Miklós Hoffmann
Institute of Mathematics and Computer Science,
Károly Eszterházy College, Eger, Hungary
e-mail: hofi@ektf.hu

Annales Mathematicae et Informaticae
36 *(2009) pp. 85–101*
http://ami.ektf.hu

Introducing general redundancy criteria for clausal tableaux, and proposing resolution tableaux

Gergely Kovásznai[a], Gábor Kusper[b]

[a]Department of Information Technology
Eszterházy Károly College, Eger, Hungary

[b]Department of Computing Science
Eszterházy Károly College, Eger, Hungary

Submitted 8 October 2008; Accepted 6 December 2009

Abstract

Hyper tableau calculi are well-known as attempts to combine hyper-resolution and tableaux. Besides their soundness and completeness, it is also important to give an appropriate redundancy criterion. The task of such a criterion is to filter out "unnecessary" clauses being attached to a given tableau. This is why we investigate what redundancy criteria can be defined for clausal tableaux, in general.

This investigation leaded us to a general idea for combining resolution calculi and tableaux. The goal is the same as in the case of hyper-tableau calculi: to split (hyper-)resolution derivations into branches. We propose a novel method called resolution tableaux. Resolution tableaux are more general than hyper tableaux, since any resolution calculus (not only hyper-resolution) can be applied, like, e.g., binary resolution, input resolution, or lock resolution etc. We prove that any resolution tableau calculus inherits the soundness and the completeness of the resolution calculus which is being applied. Hence, resolution tableaux can be regarded as a kind of parallelization of resolution.

1. Introduction

Hyper tableau calculi (e.g., hyper tableaux [2, 3], constrained hyper tableaux [6], rigid hyper tableaux [11], and hyperS tableaux [8, 9] etc.) are well-known as attempts to combine hyper-resolution and tableaux. Besides their soundness and completeness[1], it is also important to give an appropriate *redundancy criterion.* The task of such a criterion is to filter out "unnecessary" clauses being attached to a given tableau. This is why we investigate what redundancy criteria can be defined for *clausal tableaux*, in general. A clausal tableau is actually a tableau whose vertices are labeled with literals. Besides hyper tableau calculi, there are other well-known tableau calculi which apply clausal tableaux as well, like, e.g., clause tableaux [7] and connection tableaux [12]. We give a detailed investigation on appropriate redundancy criteria for clausal tableaux, and propose two possible candidates. As it will be seen, the second criterion is more general than the first one, and is also more general than the redundancy criterion for (purified) hyper tableaux [2]. We illustrate how to employ our redundancy criteria, by examples.

Investigation on redundancy leaded us to a general idea for combining resolution calculi and tableaux. The goal is the same as in the case of hyper-tableau calculi: to split (hyper-)resolution derivations into branches. First, we propose a general way of representing any resolution calculus (and illustrate it by examples), and then we introduce a novel method called *resolution tableaux.* Resolution tableaux are more general than hyper tableaux, since any resolution calculus (not only hyper-resolution) can be applied, like, e.g., binary resolution, input resolution, or lock resolution etc. We prove that any resolution tableau calculus inherits *the soundness and the completeness* of the resolution calculus which is being applied. By the use of resolution tableaux, any resolution derivation can be split into separate branches, hence resolution tableaux can be regarded as a kind of *parallelization* of resolution, as it will be illustrated by an example.

The structure of the paper is as follows. In Section 2, basic definitions and concepts are introduced. After this, let us depart from the logical order written above. First, let us propose resolution tableaux in Section 3, and then introduce our results on redundancy criteria for clausal tableaux, in Section 4. As it will be seen, the latter topic is in close connection with resolution tableaux, as detailed in Section 5.

2. Preliminaries

In the followings, we assume that the reader is familiar with the basic concepts of first-order logic. Nevertheless, let us present a few crucial concepts.

A *literal* is a formula either A or $\neg A$ where A is an atomic formula. A is classified as a positive, $\neg A$ as a negative literal.

A *clause* is a formula $L_1 \vee L_2 \vee \ldots \vee L_n$ where $n \geqslant 0$ and each L_i is a literal ($i = 1, \ldots, n$). A clause can also be regarded as the set of its literals. The *empty clause* is denoted by $\bot$.

A clause is *positive* (negative) iff it consists of solely positive (negative) literals.

Two clauses are *independent* iff there is no variable that occurs in both of them.

[1] *Constrained hyper tableaux* and *hyperS tableaux* are sound and complete in first-order logic. *Hyper tableaux* [2] without purifying substitutions are complete only in Horn logic (c.f. [3] for improvement). *Rigid hyper tableaux* have not been proven to be complete yet.

$C\sigma$ is called an *instance* of a clause C where σ is a variable substitution. $C\sigma$ is a *new instance* if σ is a variable renaming and its range consists solely of new variables.

A clause C *subsumes* a clause D iff C has an instance $C\sigma$ such that $C\sigma \subseteq D$.

Given a formula A, let $\forall A$ denote the *universal closure* of A.

As usual, $M \models A$ denotes the fact that a formula A is *satisfied by a model* M. In the case of A being open, $M \models A$ iff $M \models \forall A$.

Two formulas A and B are *equivalent* (denoted by $A \sim B$) iff for any model M: $M \models A$ iff $M \models B$.

As it is well-known, the *most general unifier* (MGU) of two atomic formulas A and B is the most general variable substitution σ such that $A\sigma = B\sigma$. Let us generalize the definition of MGUs, as follows. The MGU of (A_1, B_1), $(A_2, B_2), \ldots,$ (A_n, B_n), where all A_i and B_i are atomic formulas, is the most general variable substitution σ such that $A_i\sigma = B_i\sigma$ for all $i = 1, \ldots, n$.

Tableaux are regarded as trees whose which vertices are labeled with formulas [15, 7]. Sometimes, for the sake of briefness, we regard a tableau as the set of all its branches. Similarly, a branch is often regarded as the sequence or the set of all the vertices in the branch. Furthermore, let us introduce the following notation:

Notation 2.1. Let $\mathcal{N}$ be a vertex set from a tableau.
(1) Let $\widehat{\mathcal{N}}$ denote the conjunction of all the labels (formulas) in $\mathcal{N}$.
(2) Let$\breve{\mathcal{N}}$ denote the disjunction of all the labels (formulas) in $\mathcal{N}$.

Sometimes it is needed to regard a tableau as a sole formula. This is why we need the following definition:

Definition 2.2 (Formula Represented by a Tableau). The *formula* $\mathcal{F}(T)$ *represented* by a tableau T is defined inductively as follows:
(1) If T consists of one single vertex labeled with a formula L, then

$$\mathcal{F}(T) = L$$

(2) If T is a compound tableau, i.e., it is in the form as can be seen in Figure 1, where L is a formula and each T_i is a tableau, then

$$\mathcal{F}(T) = L \wedge \left(\bigvee_{i=1}^{n} \mathcal{F}(T_i) \right)$$

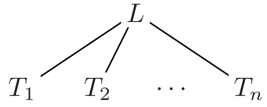


Figure 1: Compound tableau.

Let us note the following obvious fact, which says that any tableau can be regarded as the disjunction of its branches (as conjunctions).

Lemma 2.3. *For any tableau T,*

$$\mathcal{F}(T) = \bigvee_{B \in T} \widehat{B}$$

3. Resolution tableaux

The aim is to introduce a general method for combining resolution calculi and tableaux. This is why a general way of representing resolution calculi is required. We regard a resolution calculus as a *set of inference rules*, which act on clauses. Each resolution inference rule is represented as a function which can assign a clause to one or more clauses. Every time when applying such a rule, it is needed to specify a clause set (denoted by $\mathcal{I}$ and called the *input clause set*) and a sequence of clauses (denoted by $\mathbf{d}$ and called the *resolution derivation*).

Definition 3.1 (Resolution Inference Rule). A *resolution inference rule* is a function $res_{\mathcal{I},\mathbf{d}} : \mathcal{D}om \mapsto \mathcal{C}$, where

- $\mathcal{I}$ is a finite set of clauses;
- $\mathbf{d}$ is a finite sequence of clauses;
- $\mathcal{C}$ is the set of all the clauses;
- $\mathcal{D}om \subseteq P(\mathcal{C})$.

Let us illustrate by examples how well-known resolution calculi can be represented in this form. Of course, other resolution calculi could be represented in a similar way[2].

Example 3.2 (Binary Resolution). The resolution calculus, as was introduced by Robinson [13], can be represented by the set of the following resolution inference rules [1]:

(1) *Binary Resolution*:

$$binres_{\mathcal{I},\mathbf{d}}\big(\ A \vee C\ ,\ \neg B \vee D\ \big) = (C \vee D)\sigma$$

where σ is the most general unifier (MGU) of the atomic formulas A and B.

(2) *(Positive) Factoring*:

$$factor_{\mathcal{I},\mathbf{d}}\big(\ C \vee A \vee B\ \big) = (C \vee A)\sigma$$

where σ is the MGU of the atomic formulas A and B.

[2] It is to be remarked that $\mathcal{I}$ may be defined as a clause sequence (instead of a clause set) in the case of some resolution calculi, where the order of input clauses should not be neglected, like in SLD-resolution [10] and in lock resolution [4, 1, 5].

Example 3.3 (Linear Input Resolution). The linear input resolution calculus [5] can be represented by the same resolution inference rules as binary resolution, but the rule "Binary Resolution" is restricted as follows:

- one of the clauses $C \vee A$ and $D \vee \neg B$ must be the last element of $\mathbf{d}$;
- the other one must be an element of $\mathcal{I}$.

Example 3.4 (Hyper-Resolution). The (positive) hyper-resolution calculus, as was introduced by Robinson [14], can be represented by the following resolution inference rule:

$$\begin{gathered} hypres_{\mathcal{I},\mathbf{d}}(\ A_1 \vee C_1\ , \ldots,\ A_n \vee C_n\ ,\ \neg B_1 \vee \ldots \vee \neg B_n \vee D\ ) \\ \| \\ (C_1 \vee \ldots \vee C_n \vee D)\sigma \end{gathered}$$

where

- $n \geqslant 1$;
- C_i is a positive or empty clause $(i = 1, \ldots, n)$;
- D is a positive or empty clause;
- A_i and B_i are atomic formulas $(i = 1, \ldots, n)$;
- σ is the MGU of $(A_1, B_1), \ldots, (A_n, B_n)$.

When a resolution calculus (as a set of resolution inference rules) is given, an appropriate tableau can be constructed for a given input clause set $\mathcal{I}$. Such a tableau is called a *resolution tableau*, and can be constructed in a quite simple way. In every deduction steps, some clauses are to be selected, each either from a given branch of the tableau or from the input clause set $\mathcal{I}$. To the selected clauses a resolution inference rule is applied, resulting in a clause D. First D must be split into *independent subclauses*, and then these subclauses are attached to the given branch, forming distinct new branches.

Let us define resolution tableaux inductively, as follows:

Definition 3.5 (Resolution Tableaux). Let $\mathcal{R}$ be a set of resolution inference rules. Let $\mathcal{I}$ be a clause set.

(1) One single vertex labeled with $\top$ is a *resolution tableau* for $\mathcal{I}$ w.r.t. $\mathcal{R}$.

(2) – Let T be a resolution tableau for $\mathcal{I}$ w.r.t. $\mathcal{R}$.
 – Let B be a branch of T.
 – Let $C_1, \ldots, C_n$ be *new instances* of clauses in $\mathcal{I} \cup B$.
 – Let $res \in \mathcal{R}$ such that $res_{\mathcal{I},B}$ is defined on $C_1, \ldots, C_n$, and let

$$D = res_{\mathcal{I},B}(C_1, \ldots, C_n)$$

- Let $D = D_1 \vee \ldots \vee D_k$ such that each distinct D_i and D_j are *independent* clauses ($i, j = 1, \ldots, k$).[3]

The tableau that can be seen in Figure 2 is a *resolution tableau* for $\mathcal{I}$ w.r.t. $\mathcal{R}$.

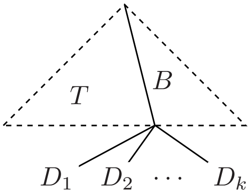


Figure 2: Attaching a clause to a branch B of a resolution tableau T.[4]

Let us point out that it is mandatory to generate *new instances* of the clauses which have been selected. Note that resolution tableau branches can be regarded (and are used) as separate resolution derivations.

A tableau calculus is regarded sound and complete in the following case: any clause set $\mathcal{I}$ is unsatisfiable iff a closed tableau exists for $\mathcal{I}$. A *closed resolution tableau* is defined as follows:

Definition 3.6 (Closed Resolution Tableaux). A resolution tableau is *closed* iff each of its branches contains $\perp$.

Assume a resolution calculus $\mathcal{R}$ which is sound and complete in first-order logic (or in a fragment of first-order logic). It is quite obvious that the resolution tableau calculus applying $\mathcal{R}$ inherits soundness and completeness. For example, since hyper-resolution is sound and complete in first-order logic, so is the hyper-resolution tableau calculus[5]. The linear input resolution tableau calculus[6] is sound and complete in Horn logic.

Theorem 3.7. *If a resolution calculus $\mathcal{R}$ is sound and complete (in a fragment of first-order logic), then so is the resolution tableau calculus applying $\mathcal{R}$.*

Proof. *Soundness:* It is to show that if there is a closed resolution tableau for $\mathcal{I}$ w.r.t. $\mathcal{R}$, then $\mathcal{I}$ is unsatisfiable.

If $\mathcal{R}$ is sound, then each inference rule $res \in \mathcal{R}$ preserves satisfiability. Let

$$res(C_1, \ldots, C_n) = D_1 \vee \ldots \vee D_k$$

where each distinct D_i and D_j are independent. It can be seen that for any model M:

[3] Furthermore, one can additionally demand that no D_i can further be split into independent subclauses ($i = 1, \ldots, k$). In this case, decomposition of clauses is unique, and can easily be solved algorithmically.

[4] New vertices labeled with $D_1, \ldots, D_k$ are attached to the leaf of B.

[5] I.e., the resolution tableau calculus applying hyper-resolution.

[6] I.e., the resolution tableau calculus applying linear input resolution.

$$\text{if } M \models C_1, \ldots, C_n, \text{ then } M \models D_1 \vee \ldots \vee D_k.$$

Because of independence:

$$\forall (D_1 \vee \ldots \vee D_k) \sim \forall D_1 \vee \ldots \vee \forall D_k$$

Summing up, for any model M:

$$\text{if } M \models C_1, \ldots, C_n, \text{ then } M \models D_1 \text{ or } M \models D_2 \text{ or } \ldots \text{ or } M \models D_k.$$

Hence, if $\mathcal{I}$ was satisfiable, then at least one branch could not be closed.

Completeness: It is to show that if $\mathcal{I}$ is unsatisfiable, then there is a closed resolution tableau for $\mathcal{I}$ w.r.t. $\mathcal{R}$. This fact is even more obvious than in the case of soundness. Since $\mathcal{R}$ is complete, there is a resolution refutation from $\mathcal{I}$. Each tableau branch can actually be regarded as a "simplified" variant of that refutation, i.e., only subclauses occuring in the refutation can occur in the branch. Since $\perp$ is deduced in the refutation and all literals of the clauses can be resolved out, obviously $\perp$ can occur in each branch. □

Note that the fact that $D_1, \ldots, D_k$ are pairwise independent has been employed only in the soundness proof.

Example 3.8 (Linear Input Resolution Tableaux). Consider the following input clause set:

$$\mathcal{I} = \left\{ \begin{array}{c} M(a, s(c), s(b)) \\ P(a) \\ M(x, x, s(x)) \vee D(y, x) \\ \neg M(x, y, z) \vee D(x, z) \\ \neg P(x) \vee \neg M(y, z, u) \vee \neg D(x, u) \vee D(x, y) \vee D(x, v) \\ \neg D(a, b) \end{array} \right\}$$

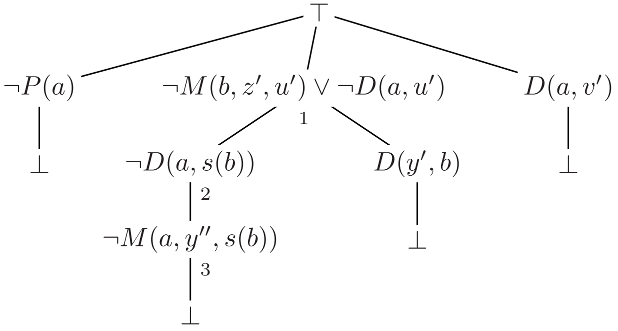
a, b, c are constants, u, v, x, y, z are variables.

In Figure 3, a closed resolution tableau for $\mathcal{I}$ w.r.t. linear input resolution (c.f. Example 3.3) can be seen.

First, the input clauses $\neg P(x) \vee \neg M(y, z, u) \vee \neg D(x, u) \vee D(x, y) \vee D(x, v)$ and $\neg D(a, b)$ are selected; the atomic formulas $D(x', y')$ and $D(a, b)$ are resolved upon by MGU $\{x'/a, y'/b\}$. As can be seen, the resolvent is split into four independent subclauses. Three branches can obviously get closed by resolving with the unit input clauses $P(a)$ and $\neg D(a, b)$.

Let us focus on the branch which contains $\neg M(b, z', u') \vee \neg D(a, u')$. Since the basis is linear input resolution, this clause (as the label of the last vertex in the branch) must be resolved with an input clause. Currently, that input clause is $M(x, x, s(x)) \vee D(y, x)$. The resolvent is split into two subclauses.

The consequent steps can be similarly performed.



Selected input clauses:

^{1}: $M(x, x, s(x)) \lor D(y, x)$
^{2}: $\neg M(x, y, z) \lor D(x, z)$
^{3}: $M(a, s(c), s(b))$

Figure 3: Closed linear input resolution tableau.

4. Redundancy criteria for clausal tableaux

Hyper tableau calculi (e.g., hyper tableaux [2, 3], rigid hyper tableaux [11], constrained hyper tableaux [6], and hyperS tableaux [8, 9]) are well-known in theorem proving. They combine hyper-resolution and tableaux. Why *hyper-resolution*? Because hyper-resolution is long known to be a key ingredient to success in theorem proving (as written in [7]). Why *tableaux*? Because of the same purpose as in resolution tableau calculi: to split hyper-resolution derivations into branches.

Hyper tableau calculi apply *clausal tableaux* as data structures representing the branches of derivations. A clausal tableau is actually a tableau whose vertices are labeled with literals. Besides hyper tableau calculi, there are other well-known tableau calculi which apply clausal tableaux as well, like, e.g., clause tableaux [7] and connection tableaux [12].

As usual, a clausal tableau is constructed by repeatedly “attaching” clauses to its branches. Clausal tableaux can be defined inductively, as follows:

Definition 4.1 (Clausal Tableaux).

(1) One single vertex labeled with $\top$ is a clausal tableau.

(2) – Let T be a clausal tableau, and B a branch of T.

– Let $E = L_1 \lor L_2 \lor \ldots \lor L_k$ be a clause.

The tableau that can be seen in Figure 4 is denoted by $T{+}^{B}E$, and is also a clausal tableau.

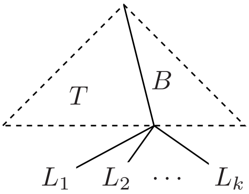


Figure 4: Attaching a clause to a branch B of a clausal tableau T.[7]

When a clause E is being attached to a branch B, the following question is essential to be answered: is it "unnecessary" to attach E to B? In other words: is E *redundant* w.r.t. B? Some clausal tableau calculi define so-called *redundancy criteria* in order to give a precise answer, by regarding only E and B (and maybe T).

Example 4.2 (Redundancy Criterion for Hyper Tableaux). Hyper tableaux have the following redundancy criterion [2]: a clause E is redundant w.r.t. a branch B iff

$$\exists L_1 \in E \text{ and } \exists L_2 \in B \text{ such that } L_1 \text{ is an instance of a new instance of } L_2.$$

We are trying to investigate what redundancy criteria can be applied in connection with clausal tableaux, in general. Besides we are extending our preceding results [8, 9], we are going to show that this problem is in close connection with resolution tableaux.

4.1. Preliminaries

First, it is essential to give a precise definition of redundancy for clausal tableaux. Note that the definition regards tableaux as formulas, as it has been introduced in Definition 2.2.

Definition 4.3 (Redundant Clause).

- Let T be a clausal tableau, and B a branch of T.
- Let E be a clause.

E is redundant w.r.t. B in T iff

$$\mathcal{F}(T) \sim \mathcal{F}\left(T+^{B}E\right)$$

Let us prove two lemmas which will be essential for proving the soundness of the redundancy criteria in the following sections.

Lemma 4.4 (Reducing Lemma).

[7] The only difference between Figure 4 and Figure 2 is that new vertices are now labeled only with literals.

- *Let T be a clausal tableau, and B a branch of T.*
- *Let M be a model.*
- *Let E be a clause.*

It holds that

$$\textit{if } M \models \mathcal{F}\left(T{+}^{B}E\right)\textit{, then } M \models \mathcal{F}\left(T\right).$$

Proof. It is to prove that $M \models \mathcal{F}\left(T{+}^{B}E\right)\theta$ implies $M \models \mathcal{F}\left(T\right)\theta$, for any valuation θ. Assume, by Lemma 2.3, we have $M \models (\widehat{B} \wedge E)\theta$ (the opposite case is obvious to prove), i.e., $M \models \widehat{B}\theta \wedge E\theta$. Thus, $M \models \widehat{B}\theta$, which implies that $M \models \mathcal{F}\left(T\right)\theta$, by Lemma 2.3. □

Lemma 4.5 (Extending Lemma).

- *Let T be a clausal tableau, and B a branch of T.*
- *Let M be a model.*
- *Let E be a clause such that $M \models E$.*

It holds that

$$\textit{if } M \models \mathcal{F}\left(T\right)\textit{, then } M \models \mathcal{F}\left(T{+}^{B}E\right).$$

Proof. It is to prove that $M \models \mathcal{F}\left(T\right)\theta$ implies $M \models \mathcal{F}\left(T{+}^{B}E\right)\theta$, for any valuation θ. Assume, by Lemma 2.3, we have $M \models \widehat{B}\theta$ (the opposite case is obvious to prove). Since $M \models E$, it holds that $M \models \widehat{B}\theta \wedge E$. Hence, $M \models (\widehat{B} \wedge E)\theta$, which implies that $M \models \mathcal{F}\left(T{+}^{B}E\right)\theta$, by Lemma 2.3. □

Note that E is an arbitrary clause in the Reducing Lemma in contrast with the Extending Lemma, where E must fulfill the following stipulation: $M \models E$.

4.2. Redundancy Criterion I

When checking if a clause E is redundant w.r.t. a tableau branch, an appropriate clause C is needed to be "extracted" from the tableau in order to "compare" C with E via instantiation. A so-called *branch clause* is a suitable candidate for this role.

Definition 4.6 (Branch Clause).

- Let T be a clausal tableau.
- Let $\mathcal{N}$ be a vertex set from T such that each branch of T contains *exactly* one element of $\mathcal{N}$.

The clause$\widecheck{\mathcal{N}}$ is a *branch clause* in T.

The Extending Lemma is to be applied in the redundancy criterion proposed later in this section. Therefore, each model of a clausal tableau T should be proven to satisfy any branch clause in T.

Lemma 4.7 (Branch Clause Lemma).

- *Let T be a clausal tableau.*
- *Let M be a model.*
- *Let C be a branch clause in T.*

It holds that

$$\textit{if } M \models \mathcal{F}(T), \textit{ then } M \models C.$$

Proof. The statement can be proven by induction, as follows:
(1) If T consists of one single vertex, then the statement obviously holds.
(2) If T is a compound tableau (cf. Figure 1), then

$$\mathcal{F}(T) \sim L \wedge \left(\bigvee_{i=1}^{n} \mathcal{F}(T_i) \right) \tag{4.1}$$

There are two cases:
(a) If $C = L$, then the statement obviously holds.
(b) Otherwise, $C = C_1 \vee \ldots \vee C_n$ where each C_i is a branch clause in T_i. By the inductive hypothesis, if $M \models \mathcal{F}(T_i)$, then $M \models C_i$, for all $i = 1, \ldots, n$. Thus,

$$\text{if } M \models \bigvee_{i=1}^{n} \mathcal{F}(T_i), \text{ then } M \models C.$$

By (4.1), the statement holds. □

The significance of the previous lemma is that a branch clause can be instantiated "without restriction", i.e., any instance (even a new instance) of a branch clause is satisfied by any model of the given tableau. Based on this fact, the following theorem on redundancy can be proven.

Theorem 4.8 (Redundancy Theorem I).

- *Let T be a clausal tableau, and B a branch of T.*
- *Let C be a branch clause in T.*
- *Let C' be a new instance of C, and E a clause such that C' subsumes E.*

It holds that

$$\mathcal{F}(T) \sim \mathcal{F}\left(T +^{B} E\right)$$

Proof. The equivalence in the right-to-left direction is a direct consequence of the Reducing Lemma. Let us prove the equivalence in the left-to-right direction, i.e., prove that for any model M:

$$\text{if } M \models \mathcal{F}(T)\text{, then } M \models \mathcal{F}\left(T +^B E\right).$$

Assume that $M \models \mathcal{F}(T)$. By Lemma 4.7, $M \models C$. Thus, $M \models C'$ also holds. So does $M \models E$. By the Extending Lemma, the proof is complete. □

Using this theorem, the following redundancy criterion can be proposed:

Definition 4.9 (Redundancy Criterion I). A clause E is redundant w.r.t. a clausal tableau T iff there is a branch clause C in T such that E is subsumed by a new instance of C.

Example 4.10 (Redundancy Criterion I). Let T be the clausal tableaux in Figure 5.

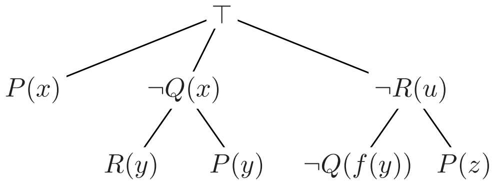


Figure 5: A clausal tableau.

When checking if a clause E is redundant w.r.t. T, Redundancy Criterion I can be applied only if E consists of at least three literals, since any branch clause instance in T consists of at least three literals (except for $\top$). For example, if

$$E = P(a) \vee R(a) \vee \neg Q(f(a))$$

then E is redundant w.r.t. T since the new branch clause instance

$$P(x') \vee R(y') \vee P(y') \vee \neg Q(f(y')) \vee P(z')$$

subsumes E, via the variable substitution $\sigma = \{x'/a, y'/a, z'/a\}$.

Note that none of all the other branch clauses subsumes E, neither $P(x) \vee \neg Q(x) \vee \neg R(u)$ nor $P(x) \vee R(y) \vee P(y) \vee \neg R(u)$ nor $P(x) \vee \neg Q(x) \vee \neg Q(f(y)) \vee P(z)$.

Note that Redundancy Criterion I is not very practicable. The problem is that a branch clause usually contains a lot of literals, and a clausal tableau may have lots of branch clauses. A more sophisticated redundancy criterion is needed.

4.3. Redundancy Criterion II

A restricted variant of a branch clause is needed, which can also be applied by the Extending Lemma. The wanted clause is a subclause of a branch clause (hence, it does not probably consist of too many literals) and does not have a common variable with a considerable part of the tableau (hence, there exist probably not too many such clauses). Let us define this clause as follows:

Definition 4.11 (Separate Branch Clause).

- Let T be a clausal tableau, and B a branch of T.
- Let $\mathcal{N}$ be a vertex set from T such that
  - each branch of T contains *at most* one element of $\mathcal{N}$;
  - B contains one element of $\mathcal{N}$;
  - there is no variable occurring both in $\mathcal{N}$ and in a branch which does not contain any element of $\mathcal{N}$.

The clause$\breve{\mathcal{N}}$ is a *separate branch clause* of B in T.

Such a clause is "separate" from the branches which do not contain it, by demanding it not to share any variable with them. As proven in the following theorem, a separate branch clause can also be used in redundancy criteria, similarly as a branch clause was.

Theorem 4.12 (Redundancy Theorem II).

- *Let T be a clausal tableau, and B a branch of T.*
- *Let C be a separate branch clause of B in T.*
- *Let C' be a new instance of C, and E a clause such that C' subsumes E.*

It holds that

$$\mathcal{F}(T) \sim \mathcal{F}\left(T +^{B} E\right)$$

Proof. Similarly as in the proof of Theorem 4.8, it is the most important to prove that for any model M:

$$\text{if } M \models \mathcal{F}(T)\text{, then } M \models \mathcal{F}\left(T +^{B} E\right)$$

The branches of T can be divided into two groups – those which contain any of the nodes included by C, and those which do not. Let t denote the set of the branches in the first group, and $\bar{t}$ the set of the ones in the second group[8]. First of all, notice the following obvious facts:

$$\mathcal{F}(T) \sim \mathcal{F}\left(\bar{t}\right) \vee \mathcal{F}(t) \tag{4.2}$$

[8] t and $\bar{t}$ are subtableaux of T. It is important that the branches of t and $\bar{t}$ are also branches in T.

$$F(T+^{B}E) \sim \mathcal{F}\left(\overline{t}\right) \vee \mathcal{F}\left(t+^{B}E\right) \tag{4.3}$$

Assume that $M \models \mathcal{F}(T)$. Since C is a branch clause in t, (4.2) and Lemma 4.7 together imply that

$$M \models \mathcal{F}\left(\overline{t}\right) \vee C$$

I.e., $M \models \forall\left(\mathcal{F}\left(\overline{t}\right) \vee C\right)$. By the assumption that $\overline{t}$ and C do not share any variable (since C is a separate branch clause), it holds that

$$M \models \forall\mathcal{F}\left(\overline{t}\right) \vee \forall C$$

Thus, there are two cases:
(1) $M \models \mathcal{F}\left(\overline{t}\right)$: It is obvious that $M \models \mathcal{F}\left(T+^{B}E\right)$, by (4.3).
(2) $M \models C$: Thus, $M \models C'$, and hence $M \models E$. Thus, $M \models \mathcal{F}\left(T+^{B}E\right)$, by the Extending Lemma. □

The redundancy criterion based on this theorem can be formulated as follows:

Definition 4.13 (Redundancy Criterion II). A clause E is redundant w.r.t. a branch B of a clausal tableau T iff there is a separate branch clause C of B in T such that E is subsumed by a new instance of C.

Example 4.14 (Redundancy Criterion II). Consider the clausal tableau T in Figure 5. Check if

$$E = P(f(a)) \vee \neg Q(f(f(a)))$$

is redundant w.r.t. T. When applying Redundancy Criterion I, the answer is no. However, by applying Redundancy Criterion II, E can be shown to be redundant w.r.t. the rightmost branch, since the new separate branch clause instance $P(z')$ subsumes E, via the variable substitution $z'/f(a)$.

Note that none of all the other separate branch clauses subsumes E, e.g., neither $P(x) \vee \neg Q(x)$ (lack of an appropriate variable substitution) nor $R(y) \vee P(y) \vee \neg Q(f(y))$. Note that E is redundant w.r.t. only the rightmost branch.

The following fact is quite interesting, and shows that some redundancy criteria in literature are specialized variants of Redundancy Criterion II.

Example 4.15 (Redundancy Criterion II for Hyper Tableaux). Let us focus on the redundancy criterion for hyper tableaux (c.f. Example 4.2, [2]). Note that it can be regarded as a special case of Redundancy Criterion II. The literal L_2 is a *separate branch clause* because of the use of *purifying substitutions* [2].

5. Conclusion

Note that Theorem 4.12 is actually applied in resolution tableaux. In a resolution tableau (however it is actually not a clausal tableau), each vertex label can be regarded as a *separate branch clause* in itself, this is why its *new instances* can be generated and used in any resolution inference step. Of course, Redundancy Criterion II (c.f. Definition 4.13) is also appropriate to be used by any resolution tableau calculus, and can be applied in a very direct way. The key is the fact that any tableau literal belongs to exactly one separate branch clause (this fact does not in general hold for clausal tableaux).

One can say that resolution tableaux have been defined according to plan. It is not accidental that vertices can be labeled not solely with literals but clauses. It is not accidental either that vertex labels are independent from each other. This solution can be regarded as the golden mean between resolution calculi and clausal tableau calculi, according to 4.12.

By the use of resolution tableaux, a resolution derivation can be split into branches, and branches (as separate derivations) can be continued simultaneously. I.e., resolution tableaux can be regarded as a kind of parallelization of resolution. Nevertheless, one can ask if it is worth to apply an additional task (namely the splitting of clauses into independent subclauses) in comparison with the advantage of shortening resolution derivations. This question could only be answered by empirical investigations.

5.1. Empirical investigations

In order to examine the practical usefulness of resolution tableaux, we implemented four different resolution calculi: *binary resolution*, *linear resolution*, *linear input resolution*, and *hyper-resolution*. Let us emphasize that purely the basic variants of those calculi have been implemented. We also implemented improved variants of the aforementioned calculi, only by applying resolution tableaux.

Then, we tested all the original and improved calculi on 1642 TPTP problems [16] (from 232 files). As it had been expectable, those calculi could not solve most of the problems in a reasonable time limit. What we primarily tried to investigate are the following questions:

- *How often* an improved calculus can solve such a problem that the original calculus cannot solve?
- If both an original calculus and its improved variant can solve a problem, *how much time* is gained by using the improved calculus?

Table 1 contains all the statistical data we have collected. Let us give an overview on the columns of the table:

- In connection with those cases when either an original calculus or its improved variant does not provide a solution (in a reasonable time limit) for the same problem, let us summarize the following data:

	Solution		**Time**		
	gained	*lost*	*gained*	*lost*	*gained/lost*
Binary	0.49%	0.59%	10.91%	29.09%	2.93%
Linear	12.3%	0%	7.45%	0.62%	626.67%
Linear input	27.57%	0%	4.48%	0%	–
Hyper	2.01%	0.97%	46.46%	18.77%	310.43%

Table 1: Empirical results.

1. **Gained answers:** The frequency of those cases when the original calculus does not provide a solution, but the improved calculus *does*.
2. **Lost answers:** The frequency of those cases when the original calculus provides a solution, but the improved calculus *does not*.

- In connection with those cases when both an original calculus and its improved variant provide a solution for the same problem, let us summarize the following data:

  1. **Gained time:** The frequency of those cases when the improved calculus provides a solution in *less time* than the original calculus.
  2. **Lost time:** The frequency of those cases when the improved calculus provides a solution in *more time* than the original calculus.
  3. **Gained time/Lost time:** We calculated the ratio of the length of the gained time to the length of the lost time in order to illustrate how it is worth to apply the improved calculus, in respect to execution time.

As it can be noticed, binary resolution tableaux do not seem very practical, in contrast with linear resolution tableaux and linear input resolution tableaux, which are absolutely worth to apply.

The conclusion in the case of hyper-resolution tableaux is quite ambiguous. Since hyper-resolution itself can be regarded as a quite powerful proof method, only in a few cases can hyper-resolution tableaux provide extra solutions. Nevertheless, the frequency of the cases when hyper-resolution tableaux shorten execution time is extremely high.

References

[1] Bachmair, L., Ganzinger, H., Resolution Theorem Proving, in: J. A. Robinson, A. Voronkov, Handbook of Automated Reasoning, Elsevier and MIT Press, North-Holland, Amsterdam, 2001, Vol. 1, Chapter 2, pp. 19–99.

[2] Baumgartner, P., Furbach, U., Niemelä, I., Hyper Tableaux, *Lecture Notes in Computer Science*, Vol. 1126, 1996, pp. 1–17.

[3] Baumgartner, P., Hyper Tableaux – The Next Generation, *Lecture Notes in Artificial Intelligence*, Vol. 1397, 1998, pp. 60–76.

[4] Boyer, R.S., Locking: A Restriction of Resolution, PhD thesis, University of Thexas at Austin, Austin, USA, 1971.

[5] Chang, C.L., Lee, R.C.T., Symbolic Logic and Mechanical Theorem Proving, Academic Press, 1973.

[6] J. van Eijck, Constrained Hyper Tableaux, *Lecture Notes in Computer Science*, Vol. 2142, 2001, pp. 232–246.

[7] Hähnle, R., Tableaux and Related Methods, in: J. A. Robinson, A. Voronkov, Handbook of Automated Reasoning, Elsevier and MIT Press, North-Holland, Amsterdam, 2001, Vol. 1, Chapter 3, pp. 100–178.

[8] Kovásznai, G., HyperS Tableaux – Heuristic Hyper Tableaux, *Acta Cybernetica*, Vol. 17, 2005, pp. 325–338.

[9] Kovásznai, G., Multi-Hyper Tableaux in Automated Theorem Proving, PhD thesis (in Hungarian), University of Debrecen, Debrecen, Hungary, 2007.

[10] Kowalski, R.A., Logic for Problem Solving, Elsevier North Holland, Amsterdam, 1979.

[11] Kühn, M., Rigid Hypertableaux, *Lecture Notes in Artificial Intelligence*, 1997, Vol. 1303, pp. 87–98.

[12] Letz, R., Schumann, J., Bayerl, S., Bibel, W., SETHEO: A High-Performance Theorem Prover, *Journal of Automated Reasoning*, Vol. 8(2), 1992, pp. 183–212.

[13] Robinson, J.A., A Machine-Oriented Logic Based on the Resolution Principle, *Journal of the ACM*, Vol. 12, 1965, pp. 23–41.

[14] Robinson, J.A., Automated Deduction with Hyper-Resolution, *International Journal of Computer Mathematics*, Vol. 1, 1965, pp. 227–234.

[15] Smullyan, R.M., First-Order Logic, Springer-Verlag, Berlin, Germany, 1968.

[16] Sutcliffe, G., The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0, *Journal of Automated Reasoning*, Vol. 43(4), 2009, pp. 337–362.

Gergely Kovásznai
Department of Information Technology
Eszterházy Károly College
P.O. Box 43
H-3301 Eger
Hungary
e-mail: `kovasz@aries.ektf.hu`

Gábor Kusper
Department of Computing Science
Eszterházy Károly College
P.O. Box 43
H-3301 Eger
Hungary
e-mail: `gkusper@aries.ektf.hu`

Annales Mathematicae et Informaticae
36 *(2009) pp. 103–110*
http://ami.ektf.hu

Localization of touching points for interpolation of discrete circles

Roland Kunkli

Department of Computer Graphics and Image Processing
University of Debrecen, Hungary

Submitted 17 February 2009; Accepted 8 May 2009

Abstract

Interpolation of an ordered set of discrete circles is discussed in this paper. By interpolation here we mean the construction of two curves which touch each of the circles and provide visually satisfactory result. Existing method frequently fail, and the crucial problem is to find good touching points on the circles. In this paper we will consider two possible solutions with pros and drawbacks.

Keywords: interpolation, circles, cyclography

MSC: 68U05

1. Introduction

Interpolation of geometric data sets is of central importance in Computer Aided Geometric Design. If geometric data consist of points, then we have standard methods to interpolate them [2, 3, 4] which give a uniquely defined fix curve. There are also recently developed methods where one can alter the shape of the interpolating curve [5, 6, 7, 8]. If, however, data set consists of other types of objects, interpolation is transferred to skinning or enveloping, and there is no unified method to solve the problem.

In this paper we address the problem of interpolation of a sequence of circles at arbitrarily given positions and radii. By interpolation we mean the construction of a pair of curves which touch the circles and the result is visually satisfactory, i.e. there are no unnecessary oscillations, bumps and loops on the curves.

This or similar problem - beside its theoretical interest - frequently arises in applications like designing tubular structures, covering problems, molecule modeling, sometimes in 3D [11], [12].

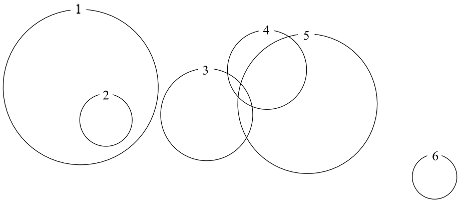


Figure 1: Position of circles where theoretically impossible to find skinning envelopes (circle No. 2 and 4)

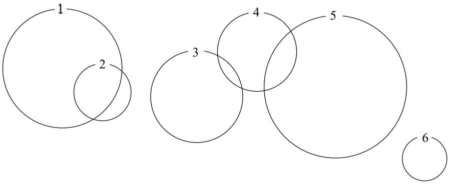


Figure 2: Position of circles where the existence of skinning envelope is theoretically possible

The sequence of circles has to satisfy a natural condition: we exclude circles which are entirely inside of other circles (c.f. Fig. 1). In Fig. 2, however we can see that small changes yield permissible positions. Otherwise the positions and radii of circles can arbitrarily be chosen.

From this point we exclude the numbering of circles from Figures - sequence starts from left to right.

Two recent approaches of the problem can be found in [9] and [10]. The first one is based on the theoretical results of envelope design of various families of curves, but as we will show in the next section, the method basically works only if one or two-parameter family of curves are given, for discrete sequences of curves the method may provide unsatisfactory result. For discrete case Slabaugh et al. provided a numerical, iterative method in [10], which works well if the radii and positions of curves do not change suddenly, that is the given data are fairly smooth. In Section 2 we will also show positions of curves for which the method simply fails, providing unnecessary oscillations and singularities.

In interpolation of circles the crucial problem is to find the touching points. In Section 3 we describe two alternative methods, each of which works well in most of the cases, even if the above mentioned methods do not work, but may fail in some extreme circumstances. Conclusions and possible further improvements close the paper.

2. Related works

Slabaugh's method is an iterative way to construct the desired curves. Let the discrete sequence of curves with centers $\mathbf{c}_i$ and radii $r_i, (i = 1, \ldots, n)$ be given. Initially pairs of Hermite arcs are defined between the consecutive circles. Considering e.g. the i^{th} and $(i+1)^{th}$ circles, two Hermite arcs are defined with touching points $\mathbf{p}_i, \mathbf{p}_{i+1}$ and tangents $\mathbf{t}_i, \mathbf{t}_{i+1}$ for the two arcs, separately. The final positions of these points and tangents are obtained by the end of the iteration steps.

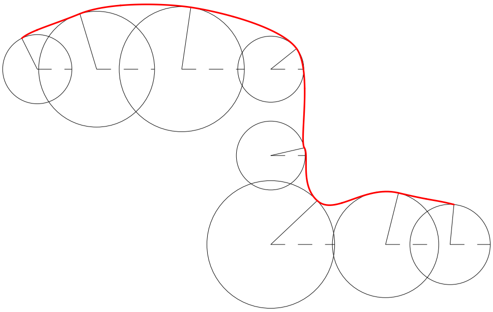


Figure 3: A good result by Slabaugh's method (from [10])

The iteration itself based on the minimization of a predefined energy function. For computational reasons the positions of the touching points and the tangents are transferred into one single variable, namely the angle α_i between the x axis and the radius pointing towards the touching point.

$$\mathbf{p}_i = \mathbf{c}_i + \begin{bmatrix} r_i \cos \alpha_i \\ r_i \sin \alpha_i \end{bmatrix},$$

$$\mathbf{t}_i = \begin{bmatrix} -k_i \sin \alpha_i \\ k_i \cos \alpha_i \end{bmatrix},$$

where k_i is a predefined constant for each circle, half of the distance between the centers $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$.

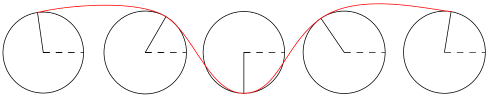


Figure 4: Automatic initial values of Slabaugh's method can yield unacceptable result even for simple data set (from [10]).

The method gives acceptable result if the sequence of curves form a "smooth" data set (c.f. Fig. 3), but even this case the initialization of the iteration, that is the starting values of the angles α_i requires user interaction, otherwise automatic values can yield obviously wrong result as one can observe in Fig. 4. A further problem is that the convergence is not proved, the number of iteration can be over 100.

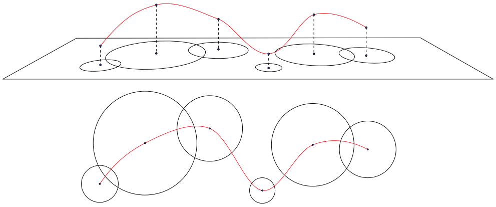


Figure 5: Peternell's method applies cyclographic mapping and spatial interpolation. Perspective and upper view of data circles and interpolation curve

Peternell's method is based on a cyclographic approach. Cyclography defines a one-to-one correspondence between the oriented circles of the plane and the spatial points by cones. This way the sequence of given circles can be transformed to a sequence of spatial points (c.f. Fig. 5). An interpolating curve through these points can be defined by any standard method and finally points of this spatial curve can be transferred back to circles on the plane by he cones. The envelope of these circles is obtained as the intersection of the plane and the envelope surface of the cones. For a more detailed description, see [9], [13].

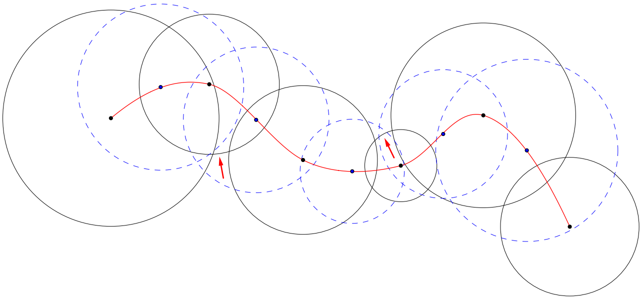


Figure 6: Classical interpolation may yields circles where the envelope cannot be constructed (positions pointed by red arrows), that is Peternell's method cannot be applied

Although this method solves the problem theoretically, it works perfectly only in the case if we know a one- or two-parameter set of circles, i.e. instead of the discrete circles $(\mathbf{c}_i, r_i)$, $i = 1, \ldots, n$, two functions $\mathbf{c}(t), r(t)$ are given.

Although these functions can be achieved from the set of discrete circles by classical interpolating methods, but this way the method not necessarily gives appropriate result, as one can observe in Fig. 6. The interpolation curve is computed by C^1 continuous Hermite arcs.

3. New methods for determining touching points

As we have learned from the previous section, the localization of possible touching points on the given circles is essential for good interpolation. In this section we discuss two methods to find these points. Although these techniques are not perfect, the second one gives acceptable results even for some extreme positions of data, for which the above mentioned methods may fail.

3.1. Planar curve driven method

The first and maybe the most natural approach is to start from a curve which interpolates the centers $\mathbf{c}_i, (i = 1, \ldots, n)$ of the given curves. For these points C^1 continuous Hermite arcs is constructed, using Bessel's method to construct the tangent vectors $\mathbf{v}_i$ at the centers $\mathbf{c}_i$.

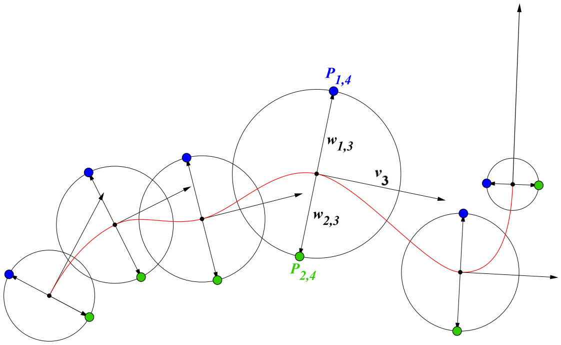


Figure 7: The planar curve driven method provides suitable touching in most cases

Now at each center consider a line orthogonal to the tangent line and consider the intersection points of this line and the corresponding circle. This method gives two points $\mathbf{p}_{1,i}, \mathbf{p}_{2,i}$ at each circle:

$$\mathbf{p}_{1,i} = \mathbf{c}_i + \frac{\mathbf{v}_{i(90^\circ)}}{|\mathbf{v}_i|} r_i,$$

$$\mathbf{p}_{2,i} = \mathbf{c}_i + \frac{\mathbf{v}_{i(-90^\circ)}}{|\mathbf{v}_i|} r_i,$$

where $\mathbf{v}_{i(\alpha)}$ denotes the vector obtained by rotating the vector $\mathbf{v}_i$ with angle α. Thus to distinguish the points on one circle, we use the orientation derived from the Hermite arcs.

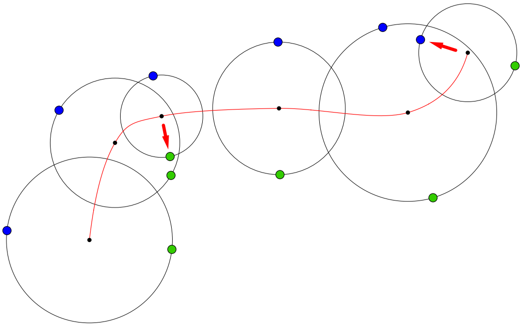


Figure 8: Sometimes computed touching points can fall into other circles

This way the two sequences of points $\mathbf{p}_{1,i}$ and $\mathbf{p}_{2,i}$ may serve us as touching points of the future interpolation curves. This method gives satisfactory result in several cases (Fig. 7), but sometimes the points $\mathbf{p}_{1,i}$ or $\mathbf{p}_{2,i}$ can fall into other circles (Fig. 8).

3.2. Spatial curve driven method

As we have seen in Section 2, circles can be transformed to spatial points by the cyclographical method. Applying this approach we can try to derive the touching point from the spatial image of the curves.

The basic idea is the following. Each circle $(\mathbf{c}_i, r_i)$ is transformed into a spatial point $\mathbf{p}_i$ and an interpolating spatial curve from Hermite arcs is constructed to these points, where the tangents are defined by Bessel's method. Now at each circle the spatial tangent line e of the Hermite arc has an intersection point T_e with the plane of the circles, from which we draw the planar tangent lines to the given circle. This way we obtain two points, E_1 and E_2, which will be touching points of the future interpolation curves (see Fig. 9).

Computationally we have to determine the angle between the line connecting the center of the circle to the intersection point of the spatial tangent, and the line connecting the center to the touching points of the planar tangents. The method works well in some extraordinary situation as well, even when the planar curve driven method would fail. This frequently happens if two neighboring circles are close to each other meanwhile their radius are strongly different (Fig. 10).

4. Conclusion and further research

Localization of touching points for interpolation of given circles are discussed. We considered two methods, which solves the problem in several cases, but none

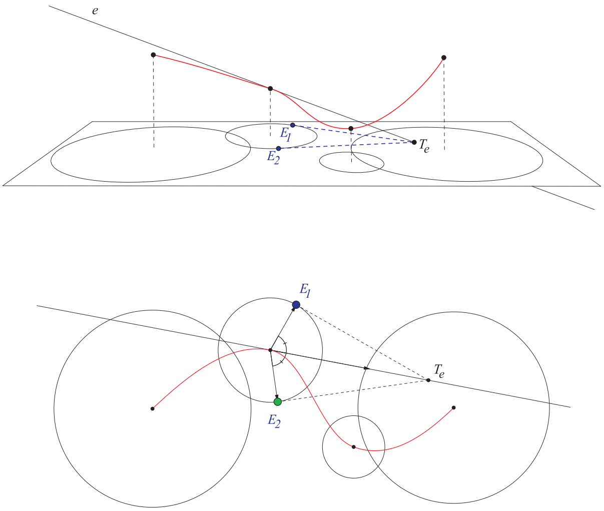


Figure 9: Planar tangent lines to the circle are drawn from the intersection point of the plane and the tangent line of the spatial curve. Perspective and upper view

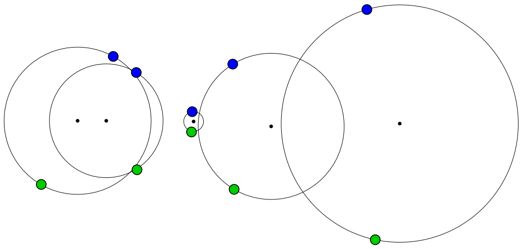


Figure 10: Spatial curve driven method gives acceptable result even in case of sudden changes of radius of neighboring circles

of them are perfect. Although the spatial curve driven method solves the problem in most cases, in extreme positions of the given circles both methods can fail, thus future improvements are required. One possibility is to change the simple Hermite interpolation by some more sophisticated methods, but we think it cannot solve the problem in general. Other geometric ways of finding touching points are under development.

References

[1] Au, C.K., Yuen, M.M.F., Unified approach to NURBS curve shape modification, *Computer-Aided Design*, 27 (1995), 85–93.

[2] Farin, G., Curves and Surface for Computer-Aided Geometric Design, 4th edition, Academic Press, New York, 1997.

[3] Piegl, L., Tiller, W., The NURBS book, Springer–Verlag, 1995.

[4] Hoschek, J., Lasser, D., Fundamentals of CAGD, AK Peters, Wellesley, MA, 1993.

[5] Tai, C.-L., Barsky, B.A., Loe, K.-F., An interpolation method with weight and relaxation parameters. In: Cohen, A., Rabut, C., Schumaker, L.L. (eds.): Curve and Surface Fitting: Saint-Malo 1999. Vanderbilt Univ. Press, Nashville, Tenessee, (2000), 393–402.

[6] Tai, C.-L., Wang, G.-J., Interpolation with slackness and continuity control and convexity-preservation using singular blending, *J. Comp. Appl. Math.*, 172 (2004), 337–361.

[7] Pan, Y.-J., Wang, G.-J., Convexity-preserving interpolation of trigonometric polynomial curves with shape parameter, *Journal of Zhejiang Univ. Sci.*, 8 (2007), 1199–1209.

[8] Hoffmann, M., Juhász, I. On interpolation by spline curves with shape parameters, *Lecture Notes in Computer Science*, 4975 (2008), 205–215.

[9] Peternell, M., Odehnal, B., Sampoli, M.L. On quadratic two-parameter families of spheres and their envelopes, *Comput. Aided Geom. Design* 25 (2008), 342–355.

[10] Slabaugh, G., Unal, G., Fang, T., Rossignac, J., Whited, B., Variational Skinning of an Ordered Set of Discrete 2D Balls, *Lecture Notes on Computer Science*, 4795 (2008), 450–461.

[11] Cheng, H., Shi, X., Quality mesh generation for molecular skin surfaces using restricted union of balls. *Proc. IEEE Visualization Conference (VIS2005)*, (2005), 399–405.

[12] Edelsbrunner, H., Deformable smooth surface design, *Discrete and Computational Geometry*, 21 (1999), 87–115.

[13] Kruithof, N., Vegter, G., Envelope Surfaces, *Proc. Annual Symposium on Computational Geometry*, (2006), 411–421.

Roland Kunkli
Department of Computer Graphics and Image Processing
University of Debrecen
H-4010 Debrecen
Hungary
e-mail: rkunkli@gmail.com

Annales Mathematicae et Informaticae
36 *(2009) pp. 111–115*
http://ami.ektf.hu

A note on integral clock triangles

Allan J. MacLeod

University of the West of Scotland

Submitted 12 September 2008; Accepted 20 January 2009

Abstract

Given two triangles with integer sides (a, b, c) and (a, b, d), and with the corresponding angles C and D such that $C \neq D$ and $C + D \neq \pi$, we show how to find a, b, c, d from any rational values of $\cos C$ and $\cos D$. For $C + D = \pi$ we show that solutions only exist for certain rational values of $\cos C$.

Keywords: elliptic curves, triangles, rational points.

MSC: 11D09, 11D25, 11Y50.

1. Introduction

In 2000, Petulante and Kaja [2] showed how to generate integer triangles with a specified value of one rational cosine. In 2007, Tengely [4] extended this to consider “clock triangle pairs” where two sides of both triangles are common. In this short note, we consider one aspect of this problem.

Let C and D be two angles of triangles, respectively with sides (a, b, c) and (a, b, d), where we suppose a and b are integers. The cosine rule implies that, if c and d are also integers, then $\cos C$ and $\cos D$ must be rational. Tengely, however, considered situations which included common angles such as $\pi/6$ and $\pi/4$. This gives sides which are possibly quadratic surds and led to the use of quadratic fields in the analysis.

The problem we consider is:

Given rational values for $\cos C$ and $\cos D$, find integer values for a, b, c, d, if possible.

To simplify the analysis, set $g = \cos C$ and $h = \cos D$. Thus we need to find integer solutions to

$$a^2 - 2gab + b^2 = c^2 \qquad a^2 - 2hab + b^2 = d^2.$$

If we define $y = c/b$ and $x = a/b$ the first equation is of the form $y^2 = x^2 - 2gx + 1$, which has the obvious solution $x = 0, y = 1$. The tangent at this point $y = 1 + Mx$ meets the curve again where $x = 2(M + g)/(1 - M^2)$. Defining $z = d/b$, the second quadratic is $z^2 = x^2 - 2hx + 1$. Substituting the x value gives

$$w^2 = M^4 + 4hM^3 + (4hg + 2)M^2 + (8g - 4h)M + (4g^2 - 4hg + 1)$$

if we define $w = z(1 - M^2)$.

The quartic, in this form, is birationally equivalent to an elliptic curve. Using the standard transformations, as described in Mordell [1], we find the elliptic curve E_{gh} is given by

$$E_{gh} : v^2 = u^3 + 2(1 - g\ h)u^2 + (g^2 - 1)(h^2 - 1)u$$

with the transformation

$$M = \frac{g(h^2 - 1) - hu + v}{(u - h^2 + 1)}$$

and thus, to find rational M and hence x, we need rational points (u, v) on these curves.

The curve E_{gh} has clearly 3 points of order 2 where $v = 0$, namely $u = 0, u = (g-1)(h+1), u = (g+1)(h-1)$, and these points are distinct if $g \neq h$. Substituting into the formula for M, gives $x = 0$ or $x = \infty$. Thus to find non-trivial solutions we must consider other points.

The relation for M has zero denominator if $u = h^2 - 1$, which we find gives a rational value $v = \pm(g - h)(h^2 - 1)$. Define the point $P = ((h^2 - 1), (g - h)(h^2 - 1)$, and the three order 2 points $T_1 = (0, 0)$, $T_2 = ((g + 1)(h - 1), 0)$ and $T_3 = ((g - 1)(h + 1), 0)$. Note that $-P = ((h^2 - 1), -(g - h)(h^2 - 1))$.

Then $P + T_1 = ((g^2 - 1), -(g - h)(g^2 - 1))$. For this to be distinct from P, we need $g^2 \neq h^2$, implying $C \neq D$ or $C \neq \pi - D$. We have $M = -g$ so $x = 0$, but using the negative of the v value gives $M = (g^2 - gh - 2)/(g + h)$ leading to $x = 4(g + h)/(4 - (g - h)^2)$. Since g and h are both in $(-1, 1)$, the denominator is always strictly positive, so if $g + h \neq 0$, the value of x is non-zero, but possibly negative, giving a solution of equation (1.1), but not real-life triangles.

As a numerical example, let $g = \cos C = 1/2$ and $h = \cos D = 1/3$, giving $x = a/b = 120/143$, suggesting $a = 120, b = 143$. This easily gives $c = 133$ and $d = 153$.

Looking at $P + T_2$ and $P + T_3$, we find trivial solutions or the above formula for x or its inverse but no new original solutions.

We now look at $2P$, which we find has $u = (g + h)^2/4$ with $v = (g + h)(4 - (g - h)^2)/8$, which gives the above formula for x. Using the negative value of v, however, leads to the following ratio

$$x = \frac{4(g + h)(g^2 + 2gh - 3h^2 + 4)(3g^2 - 2gh - h^2 - 4)}{(4 - (g - h)^2))((g - h)^2 + 4(g + h - 1))((g - h)^2 - 4(g + h + 1))}$$

which gives $x = -2441880/865007$ for $g = 1/2$ and $h = 1/3$.

A Heron triangle must have integer area, which then forces the sines of the angles to be rational. Thus we can assume that $\cos C = (1-t^2)/(1+t^2), \sin C = 2t/(1+t^2)$ and $\cos D = (1-r^2)/(1+r^2), \sin D = 2r/(1+r^2)$. For example, $t = 1/2, r = 2/3$ give $g = 3/5, h = 5/13$, which lead to $a = 260, b = 261, c = 233, d = 289$ with the two triangles having areas 27144 and 31320.

2. $C + D = \pi$

The assumption might be that, if $C + D = \pi$ or $h = -g$, then no solutions exist. This is not true - they only exist for certain g values.

Putting $h = -g$ into equation (1.3) gives

$$v^2 = u^3 + 2(g^2+1)u^2 + (g^2-1)^2 u$$

and, if we define $g = m/n$ with $0 < m < n$ coprime integers, and $X = n^2 u, Y = n^3 v$, we have

$$Y^2 = X^3 + 2(m^2+n^2)X^2 + (m^2-n^2)^2 X.$$

The roots of the right hand side show that the curve has 3 torsion points of order 2, $(0,0), (-(m-n)^2, 0), (-(m+n)^2, 0)$.

For this curve, a point (P, Q) has the X-coordinate of $2(P, Q)$ equal to

$$\frac{(P^2 - (m^2-n^2)^2)^2}{4Q^2}$$

and so, if (P, Q) is of order 4 we must have $P = \pm(m^2 - n^2)$, giving the 4 points of order 4, $(m^2-n^2, \pm 2m(m^2-n^2)), (n^2-m^2, \pm 2n(m^2-n^2))$.

Putting all of these through the various transformations we get $x = a/b$ equal to 0 or ∞.

This set of torsion points shows that the torsion subgroup is either isomorphic to $\mathbb{Z}2 \times \mathbb{Z}4$ or to $\mathbb{Z}2 \times \mathbb{Z}8$. For the latter we need points of order 8. Since $0 < m < n$, we must have

$$\frac{(P^2 - (m^2-n^2)^2)^2}{4Q^2} = n^2 - m^2 = r^2$$

and by the Nagell-Lutz theorem r will be an integer, see Silverman and Tate [3].

Thus (r, m, n) must form a primitive Pythagorean triple, and so $r = 2st, m = s^2 - t^2, n = s^2 + t^2$ for coprime integers s, t. Substituting into the above relation reduces to the quartic equation

$$(P^2 - 8st(s^2+st+t^2)P + 16s^4t^4)\,(P^2 + 8st(s^2-st+t^2)P + 16s^4t^4) = 0$$

and, investigating the discriminant of the factors, we find that, for integer roots, we must have $s^2 + t^2 = \square$. Defining $s = e^2 - f^2$ and $t = 2ef$, we find the four roots, which give the X-coordinates of the points of order 8 as

1. $X = 16f^3e(f-e)(e+f)^3$,

2. $X = 16e^3f(e-f)(e+f)^3$,
3. $X = 16f^3e(e+f)(e-f)^3$,
4. $X = 16e^3f(e+f)(f-e)^3$.

Using these X values and both the corresponding positive and negative Y values, we find that they all lead to $a/b = \pm 1$. Thus we have a solution with a and b the same length. Now we have

$$g = \frac{e^4 - 6e^2f^2 + f^4}{(e^2+f^2)^2}$$

and we find that $a = b = e^2 + f^2, c = 4ef, d = 2(e^2 - f^2)$ is a solution as long as the values are positive.

For g not of this form, to find possible solutions, we must have further rational points on the curves, which means that the rank of the curve must be greater than 0. Note that the values $g^2 - 1$ and $h^2 - 1$ giving rational points from section 1 lead to the 4 points of order 4.

Running numerical experiments gives Table 1 for small (m, n) pairs. Thus, solutions clearly do not exist for all g. For $g = 5/6$, the generator of the curve leads to the lengths $72, 35, 47, 103$.

m	n	Rank	Generator
1	2	0	
1	4	0	
3	4	1	(49,-490)
1	5	0	
4	5	0	
1	6	1	(5,90)
5	6	1	(121/4,3025/8)

Table 1: Rank of curve for small m, n.

Further experimentation, using the Birch and Swinnerton-Dyer conjecture, see Wiles [5], to estimate the heights of curve generators, shows that for $0 < m < n < 100$ the largest height is predicted to occur for $g = 30/97$, with the height being 23.8 or 47.6, depending on the height normalization used.

Using some home-grown software, the generator is found to have

$$X = \frac{701477928878^2}{4786945163^2} \qquad Y = \frac{701477928878 \times 715985634093390721663175}{4786945163^3}$$

which lead to values of a, b, c, d all having roughly 40 digits. It should be noted that John Cremona's mwrank package finds this point in seconds.

Acknowledgement. The author would like to thank the referee for several very helpful comments.

References

[1] Mordell, L.J., Diophantine Equations, *Academic Press*, New York, 1968.

[2] Petulante N., Kaja, I., How to generate all integral triangles containing a given angle, *Int. J. Math. Math. Sci.* 24 (2000), 569–572.

[3] Silverman, J.H., Tate J., Rational Points on Elliptic Curves, *Undergraduate Texts in Mathematics, Springer-Verlag*, New York, 1992.

[4] Tengely S., Triangles with two integral sides, *Ann. Math. Inform.* 34 (2007), 89–95.

[5] Wiles A., The Birch and Swinnerton-Dyer Conjecture, available at *The Clay Foundation web-site* `www.claymath.org`.

Allan J. MacLeod
Dept. of Mathematics and Statistics
University of the West of Scotland
High St., Paisley, Scotland. PA1 2BE
e-mail: `allan.macleod@uws.ac.uk`

Annales Mathematicae et Informaticae
36 *(2009) pp. 117–121*
http://ami.ektf.hu

A geometric proof to Cantor's theorem and an irrationality measure for some Cantor's series*

Diego Marques

Departamento de Matemática, Universidade de Brasília

Submitted 18 November 2008; Accepted 22 May 2009

Abstract

Generalizing a geometric idea due to J. Sondow, we give a geometric proof for the Cantor's Theorem. Moreover, it is given an irrationality measure for some Cantor series.

Keywords: Irrationality, irrationality measure, Cantor, Smarandache function.

MSC: Primary 11J72, Secondary 11J82

1. Introduction

In 2006, Jonathan Sondow gave a nice geometric proof that e is irrational. Moreover, he said that a generalization of his construction may be used to prove the Cantor's theorem. But, he did not do that in his paper, see [2]. So we give a geometric proof to Cantor's theorem using a generalization to Sondow's construction. After, it is given an irrationality measure for some Cantor series, for that we generalize the Smarandache function. Also we give an irrationality measure for e that is a bit better than the given one in [2].

2. Cantor's Theorem

Definition 2.1. Let $a_0, a_1, \dots, b_1, b_2, \dots$ be sequences of integers that satisfy the inequalities $b_n \geqslant 2$, and $0 \leqslant a_n \leqslant b_n - 1$ if $n \geqslant 1$. Then the convergent series

$$\theta := a_0 + \frac{a_1}{b_1} + \frac{a_2}{b_1 b_2} + \frac{a_3}{b_1 b_2 b_3} + \dots \tag{2.1}$$

*The author is supported by Capes.

is called *Cantor series.*

Example 2.2. The number e is a Cantor series. For see that, take $a_0 = 2, a_n = 1, b_n = n + 1$ for $n \geqslant 1$.

We recall the following theorem due to Cantor [1].

Theorem 2.3 (Cantor). *Let θ be a Cantor series. Suppose that each prime divides infinitely many of the b_n. Then θ is irrational if and only if both $a_n > 0$ and $a_n < b_n - 1$ hold infinitely often.*

Proof. For proving the necessary condition, observe that if $a_n = 0$ for $n \geqslant n_0$, then the series is a finite sum, hence θ is rational. If $a_n > 0$ infinitely often, let us to construct a nested sequence of closed intervals I_n with intersection θ. Let $I_1 = [a_0 + \frac{a_1}{b_1}, a_0 + \frac{a_1+1}{b_1}]$. Proceeding inductively, we have two possibilities, the first one, if $a_n = 0$, so define $I_n = I_{n-1}$. When $a_n \neq 0$, divide the interval I_{n-1} into $b_n - a_n + 1$ ($\geqslant 2$) subintervals, the first one with length $\frac{a_n}{b_1 \cdots b_n}$ and the other ones with equal length, namely, $\frac{1}{b_1 \cdots b_n}$, and let the first one be I_n. By construction, $|I_n| \geqslant \frac{1}{b_1 \cdots b_n}$, for all $n \in \mathbb{N}$ and when $a_n \neq 0$, the length of I_n is exactly $\frac{1}{b_1 \cdots b_n}$. By hypothesis on a_n, there exist infinitely many $n \in \mathbb{N}$, such that $|I_n| = \frac{1}{b_1 \cdots b_n}$. Thus, we have

$$I_n = \left[a_0 + \tfrac{a_1}{b_1} + \ldots + \tfrac{a_n}{b_1 \cdots b_n}, a_0 + \tfrac{a_1}{b_1} + \ldots + \tfrac{a_n+1}{b_1 \cdots b_n}\right] = \left[\tfrac{A_n}{b_1 \cdots b_n}, \tfrac{A_n+1}{b_1 \cdots b_n}\right]$$

where $A_n \in \mathbb{Z}$ for each $n \in \mathbb{N}$. Also $\theta \in I_n$ for all $n \geqslant 1$. In fact, by hypothesis it is easy see that $\theta > \frac{A_n}{b_1 \cdots b_n}$, for all $n \geqslant 1$. For the other inequality, note that $\frac{a_m}{b_m} \leqslant 1 - \frac{1}{b_m}$, for all $m \in \mathbb{N}$, therefore

$$b_1 \cdots b_n(\theta - (a_0 + \frac{a_1}{b_1} + \ldots + \frac{a_n}{b_1 \cdots b_n})) \leqslant 1. \tag{2.2}$$

Now if $a_n = b_n - 1$ for $n \geqslant n_0$, then θ is the right-hand endpoint of I_{n_0-1}, because each I_n contains that endpoint and the lengths of the I_n tend to zero. Hence again θ is rational. For showing the sufficient condition, note that if $a_m < b_m - 1$, then holds the strict inequality in (2.2), for each $n < m$. Since $a_n > 0$ holds infinitely often,

$$\bigcap_{n=1}^{\infty} I_n = \theta.$$

Suppose that $\theta = \frac{p}{q} \in \mathbb{Q}$. Each prime number divides infinitely many b_n, so there exist n_0 sufficiently large such that $q | b_1 \cdots b_{n_0}$ and $a_{n_0} \neq 0$. Hence $b_1 \cdots b_{n_0} = kq$ for some $k \in \mathbb{N}$. Take $N \geqslant n_0$, such that, $a_{N+1} < b_{N+1} - 1$. Hence θ lies in interior of I_N. Also $I_N = I_{n_0+k}$ for some $k \geqslant 0$. Suppose $I_N = I_{n_0}$. We can write $\theta = \frac{kp}{b_1 \cdots b_{n_0}}$, thus $\frac{A_{n_0}}{b_1 \cdots b_{n_0}} < \frac{kp}{b_1 \cdots b_{n_0}} < \frac{A_{n_0}+1}{b_1 \cdots b_{n_0}}$. But that is a contradiction. If $I_N = I_{n_0+k}$, for $k \geqslant 1$, then we write $\theta = \frac{kpb_{n_0+1} \cdots b_{n_0+k}}{b_1 \cdots b_{n_0+k}}$. But that is again a contradiction. Therefore, it follows the irrationality of θ. □

3. Irrationality measure

The next step is to give an irrationality measure for some Cantor series. Now, we construct an uncountable family of functions, where one of them is exactly a well-known function for us.

Definition 3.1. Given $\sigma = (b_1, b_2, \dots) \in \mathbb{N}^\infty$, satisfying

$(*)$ For all p prime number, the set $\{n \in \mathbb{N} \mid p | b_n\}$ is infinite.

We define the function $D(\cdot, \sigma) : \mathbb{Z}^* \to \mathbb{N}$, by

$$D(q, \sigma) := \min\{n \in \mathbb{N} \mid q | b_1 \cdots b_n\}$$

Note that $D(\cdot, \sigma)$ is well defined, by condition $(*)$ and the well-ordering theorem.

In [2], J. Sondow showed that for all integers p and q with $q > 1$,

$$\left| e - \frac{p}{q} \right| > \frac{1}{(S(q)+1)!}, \tag{3.1}$$

where $S(q)$ is the smallest positive integer such that $S(q)!$ is a multiple of q (the so-called Smarandache function, see [3]). Note that if $\eta = (1, 2, 3, \dots)$, then $D(q, \eta) = S(q)$. Since e is a Cantor series and $D(\cdot, \sigma)$ is a generalization of Smarandache function, it is natural to think in a generalization or an improvement to the inequality in (3.1).

Lemma 3.2. *Given $n \in \mathbb{N}$, we have*

$$\left| \theta - \frac{m}{b_1 \cdots b_n} \right| \geqslant \min\left\{ \left| \theta - \frac{A_n}{b_1 \cdots b_n} \right|, \left| \theta - \frac{A_n + 1}{b_1 \cdots b_n} \right| \right\} \tag{3.2}$$

for all $m \in \mathbb{Z}$.

Proof. Suppose that the result fail for some $m \in \mathbb{Z}$. So, $\frac{m}{b_1 \cdots b_n}$ lies in interior of I_n. Contradiction. Hence (3.2) holds for all $m \in \mathbb{Z}$. □

Proposition 3.3. *Suppose that a Cantor series θ, like in (2.1) and satisfying $(*)$, is an irrational number. For all integers $p \in \mathbb{Z}$ and $q \in \mathbb{Z}^*$, with $D(q, \sigma) > 1$, let k be the smallest integer greater than $D(q, \sigma)$ such that the interval I_k lies in the interior of $I_{D(q,\sigma)}$. Then*

$$\left| \theta - \frac{p}{q} \right| > \frac{\min\{a_k, b_k - a_k - 1\}}{b_1 \cdots b_k} \tag{3.3}$$

where $\sigma = (b_1, b_2, \dots)$.

Proof. Let $\sigma = (b_1, b_2, \dots)$. Set $n = D(q,\sigma)$ and $m = \frac{pb_1 \cdots b_n}{q}$. Therefore m and n are integers and

$$\begin{aligned} \left|\theta - \frac{p}{q}\right| &= \left|\theta - \frac{m}{b_1 \cdots b_n}\right| \\ &\geqslant \min\left\{\left|\theta - \frac{A_n}{b_1 \cdots b_n}\right|, \left|\theta - \frac{A_n + 1}{b_1 \cdots b_n}\right|\right\} \qquad (3.4) \\ &> \frac{\min\{a_k, b_k - a_k - 1\}}{b_1 \cdots b_k}. \qquad (3.5) \end{aligned}$$

The inequalities (3.4) and (3.5) follow respectively by Lemma 3.2 and the hypothesis on k. □

The result below gives a slight improvement to (3.1).

Corollary 3.3. *If p and q are integers, with $q \neq 0$, then*

$$\left|e - \frac{p}{q}\right| > \frac{1}{(D(q,\sigma) + 2)!}, \tag{3.6}$$

where $\sigma = (2, 3, 4, \dots)$.

Proof. Since that $\min_{p\in\mathbb{Z}} |e - p| > 0.28 > \frac{1}{6}$, then (3.6) holds in the case $q = \pm 1$. In case $q \neq \pm 1$ the inequality also holds by Proposition 3.3 and Example 2.2. Moreover, in this case we have $S(q) - 1 \in \{n \in \mathbb{N} \mid q|(n+1)!\}$ and $D(q,\sigma) + 1 \in \{n \in \mathbb{N} \mid q|n!\}$. Thus $S(q) = D(q,\sigma) + 1$. Hence

$$\left|e - \frac{p}{q}\right| > \frac{1}{(D(q,\sigma) + 2)!} = \frac{1}{(S(q) + 1)!}.$$

□

Actually, the improvement happens only because (3.6) also holds for $q = \pm 1$.

Example 3.4. The number $\xi := \frac{1}{(1!)^5} + \frac{1}{(2!)^5} + \frac{1}{(3!)^5} + \dots = 1.031378\dots$ is irrational, moreover for $p, q \in \mathbb{Z}$ with $q \neq 0$, we have

$$\left|\xi - \frac{p}{q}\right| > \frac{1}{(D(q,\sigma) + 2)!^5}$$

where $\sigma = (2^5, 3^5, \dots)$.

Acknowledgements. The author would like to thank Jonathan Sondow, Luiz Antônio Monte and Ana Paula Chaves for their helpful comments.

References

[1] Cantor, G., Ueber die einfachen Zahlensysteme, *Zeitschrift fur Mathematik und Physik*, 14 (1869) 121–128.

[2] Sondow, J., A geometric proof that e is irrational and a new measure of its irrationality, *Amer. Math. Monthly*, 113 (2006) 637–641.

[3] Weisstein, E, W., Smarandache function, MathWorld– A Wolfram Web Resource, published electronically at `http://mathworld.wolfram.com/SmarandacheFunction.html`.

Diego Marques
Universidade de Brasília,
Brasília, DF,
Brazil
e-mail: `diego@mat.unb.br`

Annales Mathematicae et Informaticae
36 *(2009) pp. 123–132*
http://ami.ektf.hu

An almost sure limit theorem for α-mixing random fields

Tibor Tómács

Department of Applied Mathematics
Eszterházy Károly College, Eger, Hungary

Submitted 15 September 2009; Accepted 3 November 2009

Abstract

An almost sure limit theorem with logarithmic averages for α-mixing random fields is presented.

Keywords: Almost sure limit theorem, multiindex, random field, α-mixing random field, strong law of large numbers

MSC: 60F15, 60F17

1. Introduction

Let $\mathbb{N}$ be the set of the positive integers, $\mathbb{R}$ the set of real numbers and $\mathcal{B}$ the σ-algebra of Borel sets of $\mathbb{R}$. Let δ_x be the unit mass at point x, that is $\delta_x \colon \mathcal{B} \to \mathbb{R}$, $\delta_x(B) = 1$ if $x \in B$ and $\delta_x(B) = 0$ if $x \notin B$. Denote $\xrightarrow{\mathrm{w}} \mu$ the weak convergence to the probability measure μ. In the following all random variables defined on a fixed probability space $(\Omega, \mathcal{F}, \mathrm{P})$. Almost sure (a.s.) limit theorems state that

$$\frac{1}{D_n} \sum_{k=1}^{n} d_k \delta_{\zeta_k(\omega)} \xrightarrow{\mathrm{w}} \mu \quad \text{as} \quad n \to \infty, \quad \text{for almost every} \quad \omega \in \Omega,$$

where ζ_k $(k \in \mathbb{N})$ are random variables. The simplest form of it is the so-called classical a.s. central limit theorem, in which $\zeta_k = (X_1 + \cdots + X_k)/\sqrt{k}$, where $X_1, X_2, \ldots$ are independent identically distributed (i.i.d.) random variables with expectation 0 and variance 1, moreover $d_k = 1/k$, $D_n = \log n$ and μ is the standard normal distribution $\mathcal{N}(0,1)$. (See Berkes [1] for an overview.)

Let $\mathbb{N}^d$ be the positive integer d-dimensional lattice points, where d is a fixed positive integer. In this paper $\mathbf{k} = (k_1, \ldots, k_d), \mathbf{n} = (n_1, \ldots, n_d), \ldots \in \mathbb{N}^d$. Relations $\leqslant$, $\not<$, min, max, $\to$ etc. are defined coordinatewise, i.e. $\mathbf{n} \to \infty$ means that

$n_i \to \infty$ for all $i \in \{1, \dots, d\}$. Let $|\mathbf{n}| = \prod_{i=1}^d n_i$ and $|\log \mathbf{n}| = \prod_{i=1}^d \log_+ n_i$, where $\log_+ x = \log x$ if $x \geqslant e$ and $\log_+ x = 1$ if $x < e$. The general form of the multiindex version of the a.s. limit theorems is

$$\frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \delta_{\zeta_{\mathbf{k}}(\omega)} \xrightarrow{\text{w}} \mu \quad \text{as} \quad \mathbf{n} \to \infty, \quad \text{for almost every} \quad \omega \in \Omega,$$

where $\{\zeta_{\mathbf{k}}, \mathbf{k} \in \mathbb{N}^d\}$ is a random field (multiindex sequence of random variables). In the multiindex version of the classical a.s. central limit theorem $X_{\mathbf{i}}, \mathbf{i} \in \mathbb{N}^d$ i.i.d. random variables with expectation 0 and variance 1, $\zeta_{\mathbf{k}} = \sum_{\mathbf{i} \leqslant \mathbf{k}} X_{\mathbf{i}} / \sqrt{|\mathbf{k}|}$, $d_{\mathbf{k}} = 1/|\mathbf{k}|$, $D_{\mathbf{n}} = 1/|\log \mathbf{n}|$ and $\mu = \mathcal{N}(0,1)$. It is well-known that generally the multiindex cases are not direct consequences of the corresponding theorems for ordinary sequences.

Fazekas and Rychlik proved in [5] a general a.s. limit theorem for multiindex sequences of metric space valued random elements. Tómács proved in [8] an a.s. central limit theorem for m-dependent random fields. In this paper we shall prove an a.s. limit theorem with logarithmic averages for α-mixing random fields (Theorem 2.5). Its onedimension version for $\mu = \mathcal{N}(0,1)$ is proved by Fazekas and Rychlik (see [4, Proposition 3.2]). In the proof of Theorem 2.5 we shall use a multiindex strong law of large numbers (Theorem 2.1). In the proof of Theorem 2.3 we shall follow ideas of Berkes and Csáki [2].

Throughout the paper we use the following notation. Let $\mathbb{R}_+$ be the set of the positive real numbers. If $a_1, a_2, \ldots \in \mathbb{R}$ then in case $A = \emptyset$ let $\max_{k \in A} a_k = 0$ and $\sum_{k \in A} a_k = 0$. Let $[A]$ be the closure of $A \subset \mathbb{R}$ and $\partial A = [A] \cap [\,\overline{A}\,]$.

If ξ is a random variable, then let μ_ξ denote the distribution of ξ, $\|\xi\|_\infty = \inf\{c \in \mathbb{R} : \mathrm{P}(|\xi| \leqslant c) = 1\}$ and $\sigma(\xi) = \{\xi^{-1}(B) : B \in \mathcal{B}\}$.

In the following let $\{c_k^{(i)} \in \mathbb{R}_+, k \in \mathbb{N}\}$ be increasing sequences with $c_{k+1}^{(i)}/c_k^{(i)} = O(1)$, $\lim_{n \to \infty} c_n^{(i)} = \infty$ for each $i = 1, \ldots, d$, and the sequences $\{d_k^{(i)} \in \mathbb{R}_+, k \in \mathbb{N}\}$ have the next properties: $d_k^{(i)} \leqslant \log(c_{k+1}^{(i)}/c_k^{(i)})$ for all $k \in \mathbb{N}$ and $i = 1, \ldots, d$, moreover $\sum_{k=1}^\infty d_k^{(i)} = \infty$ for each $i = 1, \ldots, d$. Let $d_{\mathbf{k}} = \prod_{i=1}^d d_{k_i}^{(i)}$, $D_{\mathbf{n}} = \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}}$ and $D_{n_i}^{(i)} = \sum_{k=1}^{n_i} d_k^{(i)}$.

2. Results

Theorem 2.1. *Let $\{\xi_{\mathbf{i}}, \mathbf{i} \in \mathbb{N}^d\}$ be a uniformly bounded random field, namely there exists $c \in \mathbb{R}_+$ such that $|\xi_{\mathbf{i}}| \leqslant c$ a.s. for all $\mathbf{i} \in \mathbb{N}^d$. Assume that there exist $c_1, c_2, \varepsilon \in \mathbb{R}_+$ and $\alpha_{\mathbf{k},\mathbf{l}} \in \mathbb{R}$ $(\mathbf{k}, \mathbf{l} \in \mathbb{N}^d)$ such that*

$$\sum_{\mathbf{l} \leqslant \mathbf{n}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} d_{\mathbf{l}} \alpha_{\mathbf{k},\mathbf{l}} \leqslant c_1 D_{\mathbf{n}}^2 \prod_{i=1}^d \left(\log D_{n_i}^{(i)} \right)^{-1-\varepsilon} \tag{2.1}$$

for all enough large $n_i \in \mathbb{N}$, and

$$|\mathrm{E}\,\xi_{\mathbf{k}}\xi_{\mathbf{l}}| \leqslant c_2 \left(\prod_{i=1}^{d} \left(\log_+ \log_+ \frac{c_{m_i}^{(i)}}{c_{h_i}^{(i)}} \right)^{-1-\varepsilon} + \alpha_{\mathbf{k},\mathbf{l}} \right) \tag{2.2}$$

for each $\mathbf{k}, \mathbf{l} \in \mathbb{N}^d$, where $\mathbf{h} = \min\{\mathbf{k}, \mathbf{l}\}$ and $\mathbf{m} = \max\{\mathbf{k}, \mathbf{l}\}$. Then

$$\frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \xi_{\mathbf{k}} \to 0 \quad \textit{as} \quad \mathbf{n} \to \infty \quad \textit{a.s.}$$

Definition 2.2. The *α-mixing coefficient* of the random variables ξ and η is

$$\alpha(\xi, \eta) = \alpha\big(\sigma(\xi), \sigma(\eta)\big) = \sup_{\substack{A \in \sigma(\xi) \\ B \in \sigma(\eta)}} |\,\mathrm{P}(AB) - \mathrm{P}(A)\,\mathrm{P}(B)|.$$

Theorem 2.3. *Let $\{\zeta_{\mathbf{k}}, \mathbf{k} \in \mathbb{N}^d\}$ be a random field. Assume that there exist random variables $\zeta_{\mathbf{h},\mathbf{l}}$ $(\mathbf{h} \leqslant \mathbf{l})$ and $c_1, c_2, c_3, \varepsilon \in \mathbb{R}_+$ such that*

$$|\zeta_{\mathbf{k}} - \zeta_{\mathbf{h},\mathbf{k}}| \geqslant c_1 \quad \textit{a.s.} \quad \forall \mathbf{h}, \mathbf{k} \in \mathbb{N}^d \quad \textit{for which} \quad \mathbf{h} \leqslant \mathbf{k}, \tag{2.3}$$

$$\mathrm{E} \min\left\{ (\zeta_{\mathbf{l}} - \zeta_{\mathbf{h},\mathbf{l}})^2, 1 \right\} \leqslant c_2 \prod_{i=1}^{d} \left(\log_+ \log_+ \frac{c_{l_i}^{(i)}}{c_{h_i}^{(i)}} \right)^{-2-2\varepsilon} \tag{2.4}$$

for all $\mathbf{h}, \mathbf{l} \in \mathbb{N}^d$ for which $\mathbf{h} \leqslant \mathbf{l}$, and

$$\sum_{\mathbf{l} \leqslant \mathbf{n}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} d_{\mathbf{l}} \alpha_{\mathbf{k},\mathbf{l}} \leqslant c_3 D_{\mathbf{n}}^2 \prod_{i=1}^{d} \left(\log D_{n_i}^{(i)} \right)^{-1-\varepsilon} \tag{2.5}$$

for all enough large $n_i \in \mathbb{N}$, where $\alpha_{\mathbf{k},\mathbf{l}} = \alpha(\zeta_{\mathbf{k}}, \zeta_{\mathbf{t},\mathbf{l}})$ with $\mathbf{t} = \min\{\mathbf{k}, \mathbf{l}\}$. Then for any probability distribution μ the following two statements are equivalent:

(1) $\displaystyle \frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \delta_{\zeta_{\mathbf{k}}(\omega)} \xrightarrow{\mathrm{w}} \mu$ *as $\mathbf{n} \to \infty$, for almost every $\omega \in \Omega$;*

(2) $\displaystyle \frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \mu_{\zeta_{\mathbf{k}}} \xrightarrow{\mathrm{w}} \mu$ *as $\mathbf{n} \to \infty$.*

Definition 2.4. The *α-mixing coefficient* of the random field $\{X_{\mathbf{n}}, \mathbf{n} \in \mathbb{N}^d\}$ is

$$\alpha(\mathbf{k}) = \sup_{\mathbf{n}} \alpha \left(\bigcup_{\mathbf{i} \leqslant \mathbf{n}} \sigma(X_{\mathbf{i}}), \bigcup_{\mathbf{i} \not\leqslant \mathbf{n}+\mathbf{k}} \sigma(X_{\mathbf{i}}) \right), \quad \mathbf{k} \in \mathbb{N}^d.$$

Theorem 2.5. *Let $\{X_{\mathbf{n}}, \mathbf{n} \in \mathbb{N}^d\}$ be an α-mixing random field with mixing coefficient*

$$\alpha(\mathbf{k}) \leqslant \frac{c}{|\log \mathbf{k}|} \tag{2.6}$$

for all $\mathbf{k} \in \mathbb{N}^d$, where $c \in \mathbb{R}_+$ is fixed. Let $S_{\mathbf{n}} = \sum_{\mathbf{k} \leqslant \mathbf{n}} X_{\mathbf{k}}$ and $\sigma_{\mathbf{n}}^2 = \mathrm{E}\, S_{\mathbf{n}}^2 > 0$. Assume that $\mathrm{E}\, X_{\mathbf{i}} = 0$ and $\mathrm{E}\, X_{\mathbf{i}}^2 < \infty$ for all $\mathbf{i} \in \mathbb{N}^d$, moreover there exist $c_1, c_2 \in \mathbb{R}_+$ and $\beta > 2/\log 2$ such that

$$|S_{\mathbf{l}}| \geqslant c_1 \sigma_{\mathbf{k}} \quad a.s. \quad \forall \mathbf{l}, \mathbf{k} \in \mathbb{N}^d \quad \text{for which} \quad \mathbf{l} \leqslant \mathbf{k} \tag{2.7}$$

and

$$\mathrm{E} \min \left\{ \frac{S_{\mathbf{r}}^2}{\sigma_{\mathbf{l}}^2}, 1 \right\} \leqslant c_2 \left(\frac{|\mathbf{h}|}{|\mathbf{l}|} \right)^{\beta} \quad \forall \mathbf{h}, \mathbf{l} \in \mathbb{N}^d \quad \text{for which} \quad \mathbf{h} \leqslant \mathbf{l}, \tag{2.8}$$

where $\mathbf{r} = 2\mathbf{h}$ if $2\mathbf{h} < \mathbf{l}$ and $\mathbf{r} = \mathbf{l}$ otherwise. If $\mu_{\zeta_{\mathbf{n}}} \xrightarrow{\mathrm{w}} \mu$ as $\mathbf{n} \to \infty$, where $\zeta_{\mathbf{n}} = S_{\mathbf{n}}/\sigma_{\mathbf{n}}$ and μ is a probability distribution, then

$$\frac{1}{|\log \mathbf{n}|} \sum_{\mathbf{k} \leqslant \mathbf{n}} \frac{1}{|\mathbf{k}|} \delta_{\zeta_{\mathbf{k}}(\omega)} \xrightarrow{\mathrm{w}} \mu \quad \text{as} \quad \mathbf{n} \to \infty, \quad \text{for almost every} \quad \omega \in \Omega.$$

3. Lemmas

You can find the proof of the next lemma in [6].

Lemma 3.1 (Covariance inequality). *If ξ and η are bounded random variables, then*

$$|\operatorname{cov}(\xi, \eta)| \leqslant 4\alpha(\xi, \eta) \|\xi\|_\infty \|\eta\|_\infty.$$

The proof of the next lemma follows from that of Theorem 11.3.3 and Corollary 11.3.4 in [3].

Lemma 3.2. *Let BL denote the set of all bounded, real-valued Lipshitz function on $\mathbb{R}$. If μ and μ_n are distributions ($n \in \mathbb{N}$), then there exists a countable set $M \subset BL$ (depending on μ) such that the following are equivalent:*

(1) $\mu_n \xrightarrow{\mathrm{w}} \mu$ *as* $n \to \infty$;

(2) $\int g \,\mathrm{d}\mu_n \to \int g \,\mathrm{d}\mu$ *as* $n \to \infty$ *for all* $g \in M$.

Lemma 3.3 (Theorem 1 of [7], p. 309). *If μ and μ_n are distributions ($n \in \mathbb{N}$), then the following are equivalent:*

(1) $\mu_n \xrightarrow{\mathrm{w}} \mu$ *as* $n \to \infty$;

(2) $\mu_n(A) \to \mu(A)$ *as* $n \to \infty$ *for all* $A \in \mathcal{B}$ *for which* $\mu(\partial A) = 0$.

Lemma 3.4. *If μ and $\mu_{\mathbf{n}}$ are distributions ($\mathbf{n} \in \mathbb{N}^d$) and $\mu_{\mathbf{n}} \xrightarrow{\mathrm{w}} \mu$ as $\mathbf{n} \to \infty$, then*

$$\frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \mu_{\mathbf{k}} \xrightarrow{\mathrm{w}} \mu \quad \text{as} \quad \mathbf{n} \to \infty.$$

Proof. By $\sum_{k_i=1}^{\infty} d_{k_i}^{(i)} = \infty$ we have

$$\frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{m} \leqslant \mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} = \prod_{i=1}^{d} \frac{\sum_{m_i \leqslant k_i \leqslant n_i} d_{k_i}^{(i)}}{\sum_{k_i \leqslant n_i} d_{k_i}^{(i)}} \to 1 \quad \text{as} \quad \mathbf{n} \to \infty \quad \forall \mathbf{m} \in \mathbb{N}^d,$$

which implies, that

$$\frac{1}{D_{\mathbf{n}}} \sum_{\substack{\mathbf{k} \leqslant \mathbf{n} \\ \mathbf{k} \ngeqslant \mathbf{m}}} d_{\mathbf{k}} = 1 - \frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{m} \leqslant \mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \to 0 \quad \text{as} \quad \mathbf{n} \to \infty \quad \forall \mathbf{m} \in \mathbb{N}^d. \tag{3.1}$$

Let $f \colon \mathbb{R} \to \mathbb{R}$ be a bounded and continuous function and $K = \sup_{x \in \mathbb{R}} |f(x)|$. Then

$$\left| \int f \, \mathrm{d}\mu_{\mathbf{n}} - \int f \, \mathrm{d}\mu \right| \leqslant \int K \, \mathrm{d}\mu_{\mathbf{n}} + \int K \, \mathrm{d}\mu = 2K, \tag{3.2}$$

moreover by $\mu_{\mathbf{n}} \xrightarrow{\mathrm{w}} \mu$ and (3.1), for any $\varepsilon > 0$ there exists $\mathbf{n}(\varepsilon) \in \mathbb{N}^d$ such that

$$\left| \int f \, \mathrm{d}\mu_{\mathbf{n}} - \int f \, \mathrm{d}\mu \right| < \frac{\varepsilon}{2} \tag{3.3}$$

and

$$\frac{1}{D_{\mathbf{n}}} \sum_{\substack{\mathbf{k} \leqslant \mathbf{n} \\ \mathbf{k} \ngeqslant \mathbf{n}(\varepsilon)}} d_{\mathbf{k}} < \frac{\varepsilon}{4K} \tag{3.4}$$

for all $\mathbf{n} \geqslant \mathbf{n}(\varepsilon)$. With notation $\gamma_{\mathbf{n}} = \frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \mu_{\mathbf{k}}$ the inequalities (3.2), (3.3) and (3.4) imply, that

$$\begin{aligned}
&\left| \int f \, \mathrm{d}\gamma_{\mathbf{n}} - \int f \, \mathrm{d}\mu \right| \leqslant \frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \left| \int f \, \mathrm{d}\mu_{\mathbf{k}} - \int f \, \mathrm{d}\mu \right| \\
&= \frac{1}{D_{\mathbf{n}}} \sum_{\substack{\mathbf{k} \leqslant \mathbf{n} \\ \mathbf{k} \ngeqslant \mathbf{n}(\varepsilon)}} d_{\mathbf{k}} \left| \int f \, \mathrm{d}\mu_{\mathbf{k}} - \int f \, \mathrm{d}\mu \right| + \frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{n}(\varepsilon) \leqslant \mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \left| \int f \, \mathrm{d}\mu_{\mathbf{k}} - \int f \, \mathrm{d}\mu \right| \\
&< \frac{1}{D_{\mathbf{n}}} \sum_{\substack{\mathbf{k} \leqslant \mathbf{n} \\ \mathbf{k} \ngeqslant \mathbf{n}(\varepsilon)}} d_{\mathbf{k}} \cdot 2K + \frac{1}{D_{\mathbf{n}}} \sum_{\mathbf{n}(\varepsilon) \leqslant \mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \cdot \frac{\varepsilon}{2} < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon
\end{aligned}$$

for all $\mathbf{n} \geqslant \mathbf{n}(\varepsilon)$. This fact implies the statement. □

4. Proof of the theorems

Proof of Theorem 2.1. By (2.2) and (2.1) we have

$$\mathrm{E} \left(\sum_{\mathbf{k} \leqslant \mathbf{n}} d_{\mathbf{k}} \xi_{\mathbf{k}} \right)^2 \leqslant \sum_{\mathbf{k} \leqslant \mathbf{n}} \sum_{\mathbf{l} \leqslant \mathbf{n}} d_{\mathbf{k}} d_{\mathbf{l}} \left| \mathrm{E} \, \xi_{\mathbf{k}} \xi_{\mathbf{l}} \right|$$

$$\leqslant c_2 \sum_{\mathbf{k}\leqslant\mathbf{n}} \sum_{\mathbf{l}\leqslant\mathbf{n}} \prod_{i=1}^{d} d_{k_i}^{(i)} d_{l_i}^{(i)} \left(\log_+ \log_+ \frac{c_{m_i}^{(i)}}{c_{h_i}^{(i)}} \right)^{-1-\varepsilon} + c_2 \sum_{\mathbf{k}\leqslant\mathbf{n}} \sum_{\mathbf{l}\leqslant\mathbf{n}} d_{\mathbf{k}} d_{\mathbf{l}} \alpha_{\mathbf{k},\mathbf{l}}$$

$$\leqslant 2c_2 \prod_{i=1}^{d} \sum_{k_i\leqslant l_i\leqslant n_i} d_{k_i}^{(i)} d_{l_i}^{(i)} \left(\log_+ \log_+ \frac{c_{l_i}^{(i)}}{c_{k_i}^{(i)}} \right)^{-1-\varepsilon} + c_2 c_1 D_{\mathbf{n}}^2 \prod_{i=1}^{d} \left(\log D_{n_i}^{(i)} \right)^{-1-\varepsilon} \tag{4.1}$$

for all enough large n_i. Now assume that $(k_i, l_i) \in A_{n_i}^{(i)}$, where

$$A_{n_i}^{(i)} = \left\{ (k_i, l_i) : k_i \leqslant l_i \leqslant n_i \text{ and } c_{l_i}^{(i)}/c_{k_i}^{(i)} \geqslant \exp\left(\sqrt{D_{n_i}^{(i)}} \right) \right\}.$$

Then $\log_+ \log_+ \left(c_{l_i}^{(i)}/c_{k_i}^{(i)} \right) \geqslant \frac{1}{2} \log D_{n_i}^{(i)}$, which implies, that

$$\sum_{(k_i,l_i)\in A_{n_i}^{(i)}} d_{k_i}^{(i)} d_{l_i}^{(i)} \left(\log_+ \log_+ \frac{c_{l_i}^{(i)}}{c_{k_i}^{(i)}} \right)^{-1-\varepsilon}$$

$$\leqslant 2^{1+\varepsilon} \left(\log D_{n_i}^{(i)} \right)^{-1-\varepsilon} \sum_{(k_i,l_i)\in A_{n_i}^{(i)}} d_{k_i}^{(i)} d_{l_i}^{(i)} \leqslant 2^{1+\varepsilon} \left(D_{n_i}^{(i)} \right)^2 \left(\log D_{n_i}^{(i)} \right)^{-1-\varepsilon}. \tag{4.2}$$

If $(k_i, l_i) \in B_{n_i}^{(i)}$, where

$$B_{n_i}^{(i)} = \left\{ (k_i, l_i) : k_i \leqslant l_i \leqslant n_i \text{ and } c_{l_i}^{(i)}/c_{k_i}^{(i)} < \exp\left(\sqrt{D_{n_i}^{(i)}} \right) \right\},$$

then with notation $M_i = \sup_k (c_{k+1}^{(i)}/c_k^{(i)})$, we get

$$\log \frac{c_{l_i+1}^{(i)}}{c_{k_i}^{(i)}} = \log \frac{c_{l_i+1}^{(i)}}{c_{l_i}^{(i)}} + \log \frac{c_{l_i}^{(i)}}{c_{k_i}^{(i)}} < \log M_i + \sqrt{D_{n_i}^{(i)}}.$$

Thus we have the following inequality, where $B_{n_i,k_i}^{(i)} = \left\{ l_i : (k_i, l_i) \in B_{n_i}^{(i)} \right\}$.

$$\sum_{(k_i,l_i)\in B_{n_i}^{(i)}} d_{k_i}^{(i)} d_{l_i}^{(i)} \left(\log_+ \log_+ \frac{c_{l_i}^{(i)}}{c_{k_i}^{(i)}} \right)^{-1-\varepsilon} \leqslant \sum_{(k_i,l_i)\in B_{n_i}^{(i)}} d_{k_i}^{(i)} d_{l_i}^{(i)}$$

$$\leqslant \sum_{(k_i,l_i)\in B_{n_i}^{(i)}} d_{k_i}^{(i)} \log \frac{c_{l_i+1}^{(i)}}{c_{l_i}^{(i)}} = \sum_{k_i=1}^{n_i} \sum_{l_i\in B_{n_i,k_i}^{(i)}} d_{k_i}^{(i)} \log \frac{c_{l_i+1}^{(i)}}{c_{l_i}^{(i)}} \leqslant \sum_{k_i=1}^{n_i} d_{k_i}^{(i)} \sum_{l_i=k_i}^{\max B_{n_i,k_i}^{(i)}} \log \frac{c_{l_i+1}^{(i)}}{c_{l_i}^{(i)}}$$

$$= \sum_{k_i=1}^{n_i} d_{k_i}^{(i)} \log \prod_{l_i=k_i}^{\max B_{n_i,k_i}^{(i)}} \frac{c_{l_i+1}^{(i)}}{c_{l_i}^{(i)}} = \sum_{k_i=1}^{n_i} d_{k_i}^{(i)} \log \frac{c_{\max B_{n_i,k_i}^{(i)}}^{(i)}}{c_{k_i}^{(i)}}$$

$$< \sum_{k_i=1}^{n_i} d_{k_i}^{(i)} \left(\log M_i + \sqrt{D_{n_i}^{(i)}} \right) \leqslant \sum_{k_i=1}^{n_i} d_{k_i}^{(i)} 2\sqrt{D_{n_i}^{(i)}} = 2 \left(D_{n_i}^{(i)} \right)^{3/2}$$

for all enough large n_i. It follows from this inequality and (4.2) that

$$\begin{aligned}&\sum_{k_i\leqslant l_i\leqslant n_i} d_{k_i}^{(i)} d_{l_i}^{(i)}\left(\log_+\log_+\frac{c_{l_i}^{(i)}}{c_{k_i}^{(i)}}\right)^{-1-\varepsilon}\\&\leqslant 2^{1+\varepsilon}\left(D_{n_i}^{(i)}\right)^2\left(\log D_{n_i}^{(i)}\right)^{-1-\varepsilon}+2\left(D_{n_i}^{(i)}\right)^{3/2}\\&\leqslant 2^{1+\varepsilon}\left(D_{n_i}^{(i)}\right)^2\left(\left(\log D_{n_i}^{(i)}\right)^{-1-\varepsilon}+\left(D_{n_i}^{(i)}\right)^{-1/2}\right)\\&\leqslant 2^{2+\varepsilon}\left(D_{n_i}^{(i)}\right)^2\left(\log D_{n_i}^{(i)}\right)^{-1-\varepsilon}\end{aligned}\tag{4.3}$$

for all enough large n_i. In the last step we use the inequality $(D_{n_i}^{(i)})^{-1/2}\leqslant(\log D_{n_i}^{(i)})^{-1-\varepsilon}$, which follows from $(D_{n_i}^{(i)})^{1/2}/(\log D_{n_i}^{(i)})^{1+\varepsilon}\to\infty$ as $n_i\to\infty$. By (4.1) and (4.3) we get

$$\mathrm{E}\left(\sum_{\mathbf{k}\leqslant\mathbf{n}} d_{\mathbf{k}}\xi_{\mathbf{k}}\right)^2\leqslant\text{const.}\prod_{i=1}^d\left(D_{n_i}^{(i)}\right)^2\left(\log D_{n_i}^{(i)}\right)^{-1-\varepsilon}\tag{4.4}$$

for all enough large n_i. Let

$$n_i(t)=\min\left\{n_i : D_{n_i}^{(i)}\leqslant\exp\left(t^{\frac{1+\varepsilon/2}{1+\varepsilon}}\right)\right\}$$

and $\mathbf{n}(\mathbf{t})=\left(n_1(t_1),\dots,n_d(t_d)\right)$. Since $n_i(t_i)\to\infty$ as $t_i\to\infty$, thus by (4.4) there exists $\mathbf{T}\in\mathbb{N}^d$, such that

$$\begin{aligned}&\mathrm{E}\sum_{\mathbf{t}\geqslant\mathbf{T}}\left(\frac{1}{D_{\mathbf{n}(\mathbf{t})}}\sum_{\mathbf{k}\leqslant\mathbf{n}(\mathbf{t})} d_{\mathbf{k}}\xi_{\mathbf{k}}\right)^2\leqslant\sum_{\mathbf{t}\geqslant\mathbf{T}}\frac{1}{D^2_{\mathbf{n}(\mathbf{t})}}\text{const.}\prod_{i=1}^d\left(D_{n_i(t_i)}^{(i)}\right)^2\left(\log D_{n_i(t_i)}^{(i)}\right)^{-1-\varepsilon}\\&\leqslant\sum_{\mathbf{t}\geqslant\mathbf{T}}\frac{1}{D^2_{\mathbf{n}(\mathbf{t})}}\text{const.}\prod_{i=1}^d\left(D_{n_i(t_i)}^{(i)}\right)^2 t_i^{-1-\varepsilon/2}=\text{const.}\prod_{i=1}^d\sum_{t_i=T_i}^{\infty} t_i^{-1-\varepsilon/2}<\infty,\end{aligned}$$

which implies

$$\frac{1}{D_{\mathbf{n}(\mathbf{t})}}\sum_{\mathbf{k}\leqslant\mathbf{n}(\mathbf{t})} d_{\mathbf{k}}\xi_{\mathbf{k}}\to 0\quad\text{as}\quad\mathbf{t}\to\infty\quad\text{a.s.}\tag{4.5}$$

For all $\mathbf{n}\in\mathbb{N}^d$ there exists $\mathbf{t}\in\mathbb{N}^d$ such that $\mathbf{n}(\mathbf{t})\leqslant\mathbf{n}\leqslant\mathbf{n}(\mathbf{t}+\mathbf{1})$, where $\mathbf{1}=(1,\dots,1)\in\mathbb{N}^d$. Thus the uniformly bounding implies

$$\left|\frac{1}{D_{\mathbf{n}}}\sum_{\mathbf{k}\leqslant\mathbf{n}} d_{\mathbf{k}}\xi_{\mathbf{k}}\right|\leqslant\left|\frac{1}{D_{\mathbf{n}(\mathbf{t})}}\sum_{\mathbf{k}\leqslant\mathbf{n}(\mathbf{t})} d_{\mathbf{k}}\xi_{\mathbf{k}}\right|+\frac{1}{D_{\mathbf{n}}}\sum_{\substack{\mathbf{k}\leqslant\mathbf{n}\\ \mathbf{k}\nleqslant\mathbf{n}(\mathbf{t})}} d_{\mathbf{k}}|\xi_{\mathbf{k}}|$$

$$\begin{aligned}&\leqslant \left|\frac{1}{D_{\mathbf{n}(\mathbf{t})}} \sum_{\mathbf{k}\leqslant\mathbf{n}(\mathbf{t})} d_{\mathbf{k}}\xi_{\mathbf{k}}\right| + \frac{1}{D_{\mathbf{n}}} \sum_{\substack{\mathbf{k}\leqslant\mathbf{n}\\ \mathbf{k}\nleqslant\mathbf{n}(\mathbf{t})}} d_{\mathbf{k}}\cdot c\\ &\leqslant \left|\frac{1}{D_{\mathbf{n}(\mathbf{t})}} \sum_{\mathbf{k}\leqslant\mathbf{n}(\mathbf{t})} d_{\mathbf{k}}\xi_{\mathbf{k}}\right| + c\left(1-\frac{D_{\mathbf{n}(\mathbf{t})}}{D_{\mathbf{n}(\mathbf{t}+\mathbf{1})}}\right) \quad \text{a.s.} \qquad (4.6)\end{aligned}$$

The reader can easy verify that $D_{\mathbf{n}(\mathbf{t})}/D_{\mathbf{n}(\mathbf{t}+\mathbf{1})} \to 1$ as $\mathbf{t}\to\infty$, so by (4.5) and (4.6) imply the statement of Theorem 2.1. □

Proof of Theorem 2.3. Let $g \in M$, where M is defined in Lemma 3.2. Then there exists $K \geqslant 1$ such that

$$|g(x)| \leqslant K \quad \text{and} \quad |g(x)-g(y)| \leqslant K|x-y| \quad \forall x,y\in\mathbb{R}. \qquad (4.7)$$

We shall prove, that with notation $\xi_{\mathbf{k}} = g(\zeta_{\mathbf{k}}) - \mathrm{E}\, g(\zeta_{\mathbf{k}})$ the conditions of Theorem 2.1 hold true. By (2.5) we get (2.1), moreover by (4.7) we have

$$|\xi_{\mathbf{k}}| \leqslant |g(\zeta_{\mathbf{k}})| + \mathrm{E}\,|g(\zeta_{\mathbf{k}})| \leqslant 2K,$$

thus $\{\xi_{\mathbf{k}}, \mathbf{k}\in\mathbb{N}^d\}$ is a uniformly bounded random field. Now we turn to (2.2). Let $\mathbf{t} = \min\{\mathbf{k},\mathbf{l}\}$. Lemma 3.1 and (4.7) imply

$$\big|\mathrm{E}\,\xi_{\mathbf{k}}\big(g(\zeta_{\mathbf{t},\mathbf{l}}) - \mathrm{E}\, g(\zeta_{\mathbf{l}})\big)\big| = \big|\mathrm{cov}\big(g(\zeta_{\mathbf{l}}), g(\zeta_{\mathbf{t},\mathbf{l}})\big)\big| \leqslant 4K^2\alpha_{\mathbf{k},\mathbf{l}}. \qquad (4.8)$$

On the other hand with notation $\eta_{\mathbf{k},\mathbf{l}} = g(\zeta_{\mathbf{l}}) - g(\zeta_{\mathbf{t},\mathbf{l}})$

$$|\mathrm{E}\,\xi_{\mathbf{k}}\eta_{\mathbf{k},\mathbf{l}}| = \big|\mathrm{cov}\big(g(\zeta_{\mathbf{k}}),\eta_{\mathbf{k},\mathbf{l}}\big)\big| \leqslant \left(\mathrm{E}\, g^2(\zeta_{\mathbf{k}})\,\mathrm{E}\,\eta^2_{\mathbf{k},\mathbf{l}}\right)^{1/2}. \qquad (4.9)$$

It is easy to see that $\big(g(x)-g(y)\big)^2 \leqslant 4K^2\min\big\{(x-y)^2, 1\big\}$, thus

$$\mathrm{E}\,\eta^2_{\mathbf{k},\mathbf{l}} \leqslant 4K^2 \min\big\{(\zeta_{\mathbf{l}} - \zeta_{\mathbf{t},\mathbf{l}})^2, 1\big\}. \qquad (4.10)$$

By (4.7) and (2.3) we have $g^2(\zeta_{\mathbf{k}}) \leqslant K^2(1+1/c_1)^2$ and

$$g^2(\zeta_{\mathbf{k}}) < K^2(c_1+1)^2 = K^2\left(1+\frac{1}{c_1}\right)^2\cdot c_1^2 \leqslant K^2\left(1+\frac{1}{c_1}\right)^2(\zeta_{\mathbf{k}} - \zeta_{\mathbf{t},\mathbf{k}})^2,$$

which imply $g^2(\zeta_{\mathbf{k}}) \leqslant \text{const.}\min\big\{(\zeta_{\mathbf{k}} - \zeta_{\mathbf{t},\mathbf{k}})^2, 1\big\}$ a.s. Using this inequality, (4.10), (4.9) and (2.4) we get the following.

$$\begin{aligned}|\mathrm{E}\,\xi_{\mathbf{k}}\eta_{\mathbf{k},\mathbf{l}}| &\leqslant \text{const.}\left(\mathrm{E}\min\big\{(\zeta_{\mathbf{k}} - \zeta_{\mathbf{t},\mathbf{k}})^2, 1\big\}\,\mathrm{E}\min\big\{(\zeta_{\mathbf{l}} - \zeta_{\mathbf{t},\mathbf{l}})^2, 1\big\}\right)^{1/2}\\ &\leqslant \text{const.}\left(\prod_{i=1}^{d} \log_+\log_+\frac{c^{(i)}_{k_i}}{c^{(i)}_{t_i}}\cdot\log_+\log_+\frac{c^{(i)}_{l_i}}{c^{(i)}_{t_i}}\right)^{-1-\varepsilon}\\ &= \text{const.}\left(\prod_{i=1}^{d}\log_+\log_+\frac{c^{(i)}_{m_i}}{c^{(i)}_{t_i}}\right)^{-1-\varepsilon}, \qquad (4.11)\end{aligned}$$

where $\mathbf{m} = \max\{\mathbf{k}, \mathbf{l}\}$. Since $|\operatorname{E}\xi_{\mathbf{k}}\xi_{\mathbf{l}}| \leqslant |\operatorname{E}\xi_{\mathbf{k}}\eta_{\mathbf{k},\mathbf{l}}| + \big|\operatorname{E}\xi_{\mathbf{k}}\big(g(\zeta_{\mathbf{t},\mathbf{l}}) - \operatorname{E} g(\zeta_{\mathbf{l}})\big)\big|$, using (4.11) and (4.8) we have (2.2). Now applying Theorem 2.1 we get

$$\frac{1}{D_{\mathbf{n}}}\sum_{\mathbf{k}\leqslant\mathbf{n}} d_{\mathbf{k}}\xi_{\mathbf{k}} \to 0 \quad \text{as} \quad \mathbf{n}\to\infty \quad \text{a.s.} \tag{4.12}$$

Let $\mu_{\mathbf{n}} = \frac{1}{D_{\mathbf{n}}}\sum_{\mathbf{k}\leqslant\mathbf{n}} d_{\mathbf{k}}\mu_{\zeta_{\mathbf{k}}}$ and $\mu_{\mathbf{n},\omega} = \frac{1}{D_{\mathbf{n}}}\sum_{\mathbf{k}\leqslant\mathbf{n}} d_{\mathbf{k}}\delta_{\zeta_{\mathbf{k}}(\omega)}$ $(\omega\in\Omega)$.

First assume that (2) is true, that is $\mu_{\mathbf{n}} \xrightarrow{\text{w}} \mu$ as $\mathbf{n}\to\infty$. Then Lemma 3.2 implies

$$\int g\,\mathrm{d}\mu_{\mathbf{n}} \to \int g\,\mathrm{d}\mu \quad \text{as} \quad \mathbf{n}\to\infty, \tag{4.13}$$

and (4.12) implies

$$\int g\,\mathrm{d}\mu_{\mathbf{n},\omega} - \int g\,\mathrm{d}\mu_{\mathbf{n}} = \frac{1}{D_{\mathbf{n}}}\sum_{\mathbf{k}\leqslant\mathbf{n}} d_{\mathbf{k}}\xi_{\mathbf{k}}(\omega) \to 0 \tag{4.14}$$

as $\mathbf{n}\to\infty$, for almost every $\omega\in\Omega$. By (4.13) and (4.14) we get $\int g\,\mathrm{d}\mu_{\mathbf{n},\omega} \to \int g\,\mathrm{d}\mu$ as $\mathbf{n}\to\infty$, for almost every $\omega\in\Omega$, thus by Lemma 3.2 we get (1).

Finally assume that (1) is true, that is $\mu_{\mathbf{n},\omega} \xrightarrow{\text{w}} \mu$ as $\mathbf{n}\to\infty$, for almost every $\omega\in\Omega$. Let $A\in\mathcal{B}$ and $\mu(\partial A) = 0$. Then by Lemma 3.3 $\mu_{\mathbf{n},\omega}(A)\to\mu(A)$ as $\mathbf{n}\to\infty$, for almost every $\omega\in\Omega$. It follows that $\mu_{\mathbf{n}}(A) = \int \mu_{\mathbf{n},\omega}(A)\,\mathrm{d}\operatorname{P}(\omega)\to\mu(A)$ as $\mathbf{n}\to\infty$. Thus using Lemma 3.3 we get (2). This completes the proof of Theorem 2.3. □

Proof of Theorem 2.5. Let $d_k^{(i)} = 1/k$, $c_k^{(i)} = k^{1/\log 2}$, $\varepsilon = (\beta\log 2 - 2)/2$, $\zeta_{\mathbf{k},\mathbf{l}} = \zeta_{\mathbf{l}} - S_{2\mathbf{k}}/\sigma_{\mathbf{l}}$ if $2\mathbf{k} < \mathbf{l}$ and $\zeta_{\mathbf{k},\mathbf{l}} = 0$ if $\mathbf{k}\leqslant\mathbf{l}$ and $2\mathbf{k}\not<\mathbf{l}$. We shall prove that conditions of Theorem 2.3 hold. It is easy to see that $\alpha_{\mathbf{k},\mathbf{l}} \leqslant \alpha(\mathbf{k})$ for all $\mathbf{k},\mathbf{l}\in\mathbb{N}^d$, where $\alpha_{\mathbf{k},\mathbf{l}}$ is defined in Theorem 2.3. Therefore by (2.6) we have

$$\begin{aligned}\sum_{\mathbf{l}\leqslant\mathbf{n}}\sum_{\mathbf{k}\leqslant\mathbf{n}} d_{\mathbf{k}}d_{\mathbf{l}}\alpha_{\mathbf{k},\mathbf{l}} &\leqslant \sum_{\mathbf{l}\leqslant\mathbf{n}}\sum_{\mathbf{k}\leqslant\mathbf{n}} \frac{c}{|\mathbf{k}|\cdot|\mathbf{l}|\cdot|\log\mathbf{k}|}\\ &= c\prod_{i=1}^{d}\left(\sum_{k=1}^{n_i}\frac{1}{k\log_+ k}\right)\left(\sum_{l=1}^{n_i}\frac{1}{l}\right).\end{aligned}\tag{4.15}$$

It is well-known that $\sum_{k=1}^{n}\frac{1}{k}\sim\log n$ and $\sum_{k=1}^{n}\frac{1}{k\log_+ k}\sim\log\log n$, where $a_n\sim b_n$ iff $\lim_{n\to\infty} a_n/b_n = 1$. So by (4.15) we have

$$\begin{aligned}\sum_{\mathbf{l}\leqslant\mathbf{n}}\sum_{\mathbf{k}\leqslant\mathbf{n}} d_{\mathbf{k}}d_{\mathbf{l}}\alpha_{\mathbf{k},\mathbf{l}} &\leqslant \text{const.}\prod_{i=1}^{d}\log\log n_i\cdot\log n_i \leqslant \text{const.}\prod_{i=1}^{d}(\log n_i)^2(\log\log n_i)^{-1-\varepsilon}\\ &\leqslant \text{const.}\prod_{i=1}^{d}(\log n_i)^2(\log D_{n_i}^{(i)})^{-1-\varepsilon} \leqslant \text{const.}D_{\mathbf{n}}^2\prod_{i=1}^{d}(\log D_{n_i}^{(i)})^{-1-\varepsilon}\end{aligned}$$

for all enough large n_i, which implies (2.5). Using (2.8)

$$\mathrm{E}\min\left\{(\zeta_{\mathbf{l}}-\zeta_{\mathbf{h},\mathbf{l}})^2,1\right\}=\mathrm{E}\min\left\{S_{\mathbf{r}}^2/\sigma_{\mathbf{l}}^2,1\right\}\leqslant \text{const.}\prod_{i=1}^{d}\left(\log_+\log_+\frac{c_{l_i}^{(i)}}{c_{h_i}^{(i)}}\right)^{-2-2\varepsilon}$$

for all $\mathbf{h},\mathbf{l}\in\mathbb{N}^d$ for which $\mathbf{h}\leqslant\mathbf{l}$, where $\mathbf{r}=2\mathbf{h}$ if $2\mathbf{h}<\mathbf{l}$ and $\mathbf{r}=\mathbf{l}$ if $\mathbf{h}\leqslant\mathbf{l}$ and $2\mathbf{h}\not<\mathbf{l}$, so we get (2.4). The reader can readily verify that (2.3) is hold as well. Now applying Lemma 3.4 and Theorem 2.3, we have

$$\frac{1}{\sum_{\mathbf{k}\leqslant\mathbf{n}}\frac{1}{|\mathbf{k}|}}\sum_{\mathbf{k}\leqslant\mathbf{n}}\frac{1}{|\mathbf{k}|}\delta_{\zeta_{\mathbf{k}}(\omega)}\xrightarrow{\text{w}}\mu \quad\text{as}\quad \mathbf{n}\to\infty,\quad\text{for almost every}\quad \omega\in\Omega.$$

Since $\sum_{\mathbf{k}\leqslant\mathbf{n}}\frac{1}{|\mathbf{k}|}\sim|\log\mathbf{n}|$, we get the statement. □

References

[1] Berkes, I., Results and problems related to the pointwise central limit theorem, In: Szyszkowicz, B. (Ed.) Asymptotic results in probability and statistics, *Elsevier, Amsterdam,* (1998), 59–96.

[2] Berkes, I., Csáki, E., A universal result in almost sure central limit theory, *Stoch. Proc. Appl.*, 94(1) (2001), 105–134.

[3] Dudley, R.M., Real Analysis and Probability, *Cambridge University Press*, (2002).

[4] Fazekas, I., Rychlik, Z., Almost sure functional limit theorems, *Annales Universitatis Mariae Curie-Skłodowska Lublin*, Vol. LVI, 1, Sectio A, (2002) 1–18.

[5] Fazekas, I., Rychlik, Z., Almost sure central limit theorems for random fields, *Math. Nachr.*, 259, (2003), 12–18.

[6] Lin, Z., Lu, C., Limit theory for mixing dependent random variables, *Science Press, New York–Beijing and Kluwer, Dordrecht–Boston–London* (1996).

[7] Shiryayev, A.N., Probability, *Springer-Verlag New York Inc.* (1984).

[8] Tómács, T., Almost sure central limit theorems for m-dependent random fields, *Acta Acad. Paed. Agriensis, Sectio Mathematicae*, 29 (2002) 89–94.

Tibor Tómács
Department of Applied Mathematics
Eszterházy Károly College
P.O. Box 43
H-3301 Eger
Hungary
e-mail: tomacs@ektf.hu

Annales Mathematicae et Informaticae
36 *(2009) pp. 133–141*
http://ami.ektf.hu

Limit theorems for the longest run

József Túri

University of Miskolc, Department of Descriptive Geometry

Submitted 21 June 2009; Accepted 5 October 2009

Abstract

Limit theorems for the longest run in a coin tossing experiment are obtained.

Keywords: almost sure limit theorem for longest run.

MSC: 60F05 Central limit and other weak theorems, 60F15 Strong theorems.

1. Introduction

Problems connected to the longest head run in a coin tossing experiment have been investigated for a long time. Erdős and Rényi (1970) proved for a fair coin that for arbitrary $0 < c_1 < 1 < c_2 < \infty$ and for almost all $\omega \in \Omega$ there exists a finite $N_0 = N_0(\omega, c_1, c_2)$ such that $[c_1 \operatorname{Log} N] \leqslant \mu(N) \leqslant [c_2 \operatorname{Log} N]$ if $N \geqslant N_0$ (here $\mu(N)$ denotes the length of the longest head run during the first N experiments, $[.]$ denotes the integer part, Log means logarithm of base 2). Erdős and Révész (1975) improved the above upper and lower bounds, moreover they proved other strong theorems for $\mu(N)$. Deheuvels (1985) Theorem 2 offers a.s. upper and lover bounds for the k-th longest head run for a biased coin. Földes (1979) studied the case of a fair coin and obtained limit theorems for the longest head run containing at most T tails. Binswanger and Embrechts (1994) gave a review of the results on the longest head run and their applications to gambling and finance. In the point of view of applications recursive algorithms for the distribution of the longest head run are important (see Kopociński (1991), Muselli (2000)). Fazekas and Noszály (2007) studied the limit distribution of the longest T-interrupted run of heads and recursive algorithms for the distribution in the case of a biased coin. Schilling (1990) gave an overview of limit theorems, algorithms and applications. Schilling (1990) studied pure head runs and runs of pure head or pure tails, too.

In this paper we study a coin tossing experiment. That is the underlying random variables are $X_1, X_2, \ldots$. We assume that $X_1, X_2, \ldots$ are independent and identically distributed with $\mathbb{P}(X_i = 1) = p$, $\mathbb{P}(X_i = 0) = q = 1 - p$. I.e. we write 1 for a head and 0 for a tail. In Section 2 we study pure runs, i.e. runs containing only heads or containing only tails. In Section 2 we prove limit theorems for the longest run. Our theorems 2.5–2.8 are versions of theorems 1–4 in Földes [9]. These are limit theorems for a fair coin. We consider the case of a biased coin in theorems 2.8 and 2.10.

Recently several papers are devoted to the study of almost sure limit theorems (see Berkes, Csáki (2001), Berkes, Dehling and Móri (1991), Fazekas and Rychlik (2002), Major (1998) and the references therein).

In Section 3 we obtain an almost sure limit theorem for the longest run (Theorem 3.1). We remark that for the longest run there is no limiting distribution (in Theorem 2.10 we give an accompanying sequence for it). However, for the logarithmic average we obtain limiting distribution. Our Theorem 3.1 is a version of Corollary 5.1 of Móri [16].

2. Limit theorems for longest runs

Consider N tossings of a coin. In this part we prove some limit theorems for longest runs. The theorems concern arbitrary pure runs (i.e. pure head runs or pure tail runs).

We shall use the next notation. Let $\xi(n, N) = \xi(n, N, \omega)$ denote the number of pure head sequences having length n.

Let $\xi^*(n, N) = \xi(n, N, \omega)$ denote the number of pure head or pure tail sequences having length n.

Let $\tilde{\xi}(n, N) = \tilde{\xi}(n, N, \omega)$ denote the number of disjoint pure head sequences with length being at least n.

Let $\tilde{\xi}^*(n, N) = \tilde{\xi}^*(n, N, \omega)$ denote the number of disjoint pure head or pure tail sequences with length being at least n.

Let $\tau(n) = \tau(n, \omega)$ denote the smallest number of casts which are necessary to get at least one pure head run of length n, that is

$$\tau(n) = \min\{N \mid \xi(n, N) > 0\}.$$

Let $\tau^*(n) = \tau^*(n, \omega)$ denote the smallest number of casts which are necessary to get at least one pure head or one pure tail run of length n, that is

$$\tau^*(n) = \min\{N \mid \xi^*(n, N) > 0\}.$$

Let $\mu(N) = \mu(N, \omega)$ denote the length of the longest pure head run in the first N trials, that is

$$\mu(N) = \max\{n \mid \xi(n, N) > 0\}.$$

Let $\mu^*(N) = \mu^*(N, \omega)$ denote the length of the longest pure head or pure tail run in the first N trials, that is

$$\mu^*(N) = \max\{n \mid \xi^*(n, N) > 0\}.$$

Here $\omega \in \Omega$, where $(\Omega, A, \mathbb{P})$ is the underlying probability space.

In this section we obtain analogues of Theorems $1-4$ in Földes [9] for arbitrary pure runs.

First consider the case of a fair coin. For convenience we quote the results of Földes.

Theorem 2.1 (Theorem 1 in [9]). *If $N \to \infty$ and $n \to \infty$ such that*

$$\frac{N}{2^{n+1}} \to \lambda > 0, \tag{2.1}$$

then we hawe

$$\lim_{N\to\infty} \mathbb{P}(\tilde{\xi}(n,N) = k) = \frac{e^{-\lambda}\lambda^k}{k!}, \quad k = 0,1,2,\dots . \tag{2.2}$$

Theorem 2.2 (Theorem 2 in [9]). *Under the condition of Theorem 2.1 the distribution of $\xi(n,N)$ converges to a compound Poisson distribution, namely*

$$\mathbb{E}(z^{\xi(n,N)}) \to \exp\left(\lambda\left(\frac{(1-\frac{1}{2})z}{1-\frac{1}{2}z} - 1\right)\right). \tag{2.3}$$

Theorem 2.3 (Theorem 3 in [9]). *For $0 < x < \infty$*

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{\tau(n)}{2^{n+1}} \leqslant x\right) = 1 - e^{-x}. \tag{2.4}$$

Theorem 2.4 (Theorem 4 in [9]). *For any integer k we have*

$$\mathbb{P}(\mu(N) - [\operatorname{Log} N] < k) = \exp(-2^{-(k+1-\{\operatorname{Log} N\})}) + o(1) \tag{2.5}$$

where $[a]$ denotes the integer part of a and $\{a\} = a - [a]$.

We use the next connection between the pure head runs and pure runs (see, for example, Schilling in [21]).

Remark 2.5. The next relation is true.

$$2\,\mathrm{card}\{\tilde{\xi}(n-1,N-1) = k\} = \mathrm{card}\{\tilde{\xi}^*(n,N) = k\}, \quad k = 0,1,2,\dots . \tag{2.6}$$

Theorem 2.6. *If $N \to \infty$ and $n \to \infty$ such that*

$$\frac{N}{2^{n+1}} \to \lambda > 0, \tag{2.7}$$

then we hawe

$$\lim_{N\to\infty} \mathbb{P}(\tilde{\xi}^*(n,N) = k) = \frac{e^{-2\lambda}(2\lambda)^k}{k!}, \quad k = 0,1,2,\dots . \tag{2.8}$$

Proof. If we use the (2.6.) connection we have for $k = 0, 1, 2, \dots$

$$\mathbb{P}(\tilde{\xi}^*(n, N) = k) = \frac{\text{card}\{\tilde{\xi}^*(n, N) = k\}}{2^N} = \frac{2\,\text{card}\{\tilde{\xi}(n-1, N-1) = k\}}{2^N} = \\ = \mathbb{P}(\tilde{\xi}(n-1, N-1) = k).$$

If $\frac{N}{2^{n+1}} \to \lambda$, then

$$\frac{N-1}{2^{(n-1)+1}} = 2\frac{N-1}{N}\frac{N}{2^{n+1}} \to 2\lambda.$$

By Theorem 2.1, we obtain that

$$\lim_{n\to\infty} \mathbb{P}(\tilde{\xi}^*(n, N) = k) = e^{-2\lambda}\frac{(2\lambda)^k}{k!}.$$

This completes the proof of Theorem 2.5. □

Theorem 2.7. *Under the condition (2.1) the distribution of* $\xi^*(n, N)$ *converges to a compound Poisson distribution, namely*

$$\lim_{N\to\infty} \mathbb{E}(z^{\xi^*(n,N)}) = \exp\left(2\lambda\left(\frac{(1-\frac{1}{2})z}{1-\frac{1}{2}z} - 1\right)\right). \tag{2.9}$$

Proof. By (2.6.), we have

$$\mathbb{E}z^{\xi^*(n,N)} = \sum_{k=0}^{\infty} z^k \mathbb{P}(\xi^*(n, N) = k) = \sum_{k=0}^{\infty} z^k\,\text{card}\{\xi^*(n, N) = k\}/2^N = \\ = \sum_{k=0}^{\infty} z^k 2\,\text{card}\{\xi(n-1, N-1) = k\}/2^N = \sum_{k=0}^{\infty} z^k \mathbb{P}(\xi(n-1, N-1) = k) = \\ = \mathbb{E}z^{\xi(n-1,N-1)}.$$

By Theorem 2.2

$$\mathbb{E}z^{\xi(n-1,N-1)} = \exp\left(2\lambda\left(\frac{(1-\frac{1}{2})z}{1-\frac{1}{2}z} - 1\right)\right).$$

This completes the proof of Theorem 2.6. □

The next theorem state that the limit distribution of $\frac{\tau^*}{2^{n+1}}$ is exponential with parameter 2.

Theorem 2.8. *For* $0 < x < \infty$

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{\tau^*(n)}{2^{n+1}} \leqslant x\right) = 1 - e^{-2x} \tag{2.10}$$

Proof. The theorem is the consequence of the calculation below and Theorem 2.3.

$$\begin{aligned}
&\mathbb{P}\left(\frac{\tau^*(n)}{2^n} > x\right) = \mathbb{P}(\text{ from } [2^n x] \text{ trials there is no run of lenght } n) = \\
&= \frac{\text{card}\{\text{ from } [2^n x] \text{ trials there is no run of lenght } n\}}{2^{[2^n x]}} = \\
&= \frac{2\ \text{card}\{\text{from } [2^n x]-1 \text{ trials there is no head run of lenght } n-1\ )}{2^{[2^n x]}} = \\
&= \mathbb{P}(\tau(n-1) > [2^n x]-1) = \mathbb{P}\left(\frac{\tau(n-1)}{2^n} > \frac{[2^n x]-1}{2^n}\right) = \\
&= \mathbb{P}\left(\frac{\tau(n-1)}{2^n} > x + a_n\right) = \mathbb{P}\left(\frac{\tau(n-1)}{2^n} - a_n > x\right)
\end{aligned}$$

where $\frac{[2^n x]-1}{2^n} = x + a_n$ and $a_n \to 0$. If we use Slutsky's theorem and Theorem 2.3, we get that

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{\tau(n-1)}{2^n} - a_n > x\right) = e^{-x}.$$

So

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{\tau^*(n)}{2^{n+1}} \leqslant x\right) = 1 - e^{-2x}.$$

This completes the proof of Theorem 2.7. □

Theorem 2.9. *For any integer k we have*

$$\mathbb{P}(\mu^*(N) - [\text{Log}(N-1)] < k) = \exp(-2^{-(k-\{\text{Log}(N-1)\})}) + o(1). \tag{2.11}$$

Proof. By Remark 2.1, we have

$$\begin{aligned}
&\mathbb{P}(\mu^*(N) - [\text{Log}(N-1)] < k) = \\
&= \frac{\text{card}\{\mu^*(N) - [\text{Log}(N-1)] < k\}}{2^N} = \\
&= 2\,\text{card}\{\mu(N-1) - [\text{Log}(N-1)] < k-1\}/2^N = \\
&= \mathbb{P}(\mu(N-1) - [\text{Log}(N-1)] < k) = \\
&= \exp\left(-2^{-(k-\{\text{Log}(N-1)\})}\right) + o(1),
\end{aligned}$$

where we applied Theorem 2.4. This completes the proof of Theorem 2.8. □

Now consider the case of a biased coin. Let p be the probability of tail and $q = 1-p$ the probability of head. Let $V_N(p)$ denote the probability that the longest run in N trials is formed by heads. Then, by Theorem 5 of Musselli [19],

$$\lim_{N\to\infty} V_N(p) = \begin{cases} 0 & \text{if } 0 \leqslant p < \frac{1}{2} \\ 1 & \text{if } \frac{1}{2} < p \leqslant 1. \end{cases} \tag{2.12}$$

Theorem 2.10. *Let $p > q$. For $0 < x < \infty$*

$$\lim_{n\to\infty} \mathbb{P}(\tau^*(n)qp^n \leqslant x) = 1 - e^{-x}. \tag{2.13}$$

Proof. We have

$$\lim_{n\to\infty} \mathbb{P}(\tau(n)qp^n \leqslant x) = 1 - e^{-x}. \tag{2.14}$$

(2.14) is mentioned in Móri [16] without proof and it is proved in Fazekas-Noszály [8]. Using (2.12), (2.14) implies (2.13). □

Theorem 2.11. *Let $p > q$. Let* Log *denote the logarithm of base $1/p$. Then for any integer k*

$$\mathbb{P}(\mu^*(N) - [\operatorname{Log} N] < k) = \exp(-qp^{k-\{\operatorname{Log} N\}}) + o(1). \tag{2.15}$$

Proof. By Gordon-Schilling-Waterman [11] or Fazekas-Noszály [8],

$$\mathbb{P}(\mu(N) - [\operatorname{Log} N] < k) = \exp(-qp^{k-\{\operatorname{Log} N\}}) + o(1). \tag{2.16}$$

(2.12) and (2.16) implies (2.15). □

3. An a.s. limit theorem for the longest run

In this part we prove an a.s. limit theorem for the longest run. Our theorem is a version of the following result of Móri. Let p be the probability of the head. Let Log denote the logarithm of base $1/p$. Let log denote the logarithm of base e.

Remark 3.1 (A particular case of Corollary 5.1 in Móri [16]).

$$\lim_{n\to\infty} \frac{1}{\log n} \sum_{i=1}^{n} \frac{1}{i} \mathbb{I}(\mu(i) - \operatorname{Log} i < t) = \int_t^{t+1} \exp(-qp^z)dz \quad \text{a.s.} \tag{3.1}$$

Let us abbreviate $\mathbb{E}(\tau^*(n))$ by $E(n)$ and $\mathbb{P}(\tau^*(n) = n)$ by $p(n)$. To prove the a.s. limit theorem for the longest run we shall need the next results.

Remark 3.2 (See Lemma 2.2 in Móri [16]).

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{\tau^*(n)}{E(n)} > t\right) = e^{-t} \tag{3.2}$$

uniformly in $t \geqslant 0$.

Proposition 3.3 (A particular case of Theorem 3.1 in Móri [16]). *Suppose that f is a positive, increasing, differentiable function such that $E(m) \sim f(m)$ and the limit*

$$c = \lim_{t\to\infty} (\log f(t))' \tag{3.3}$$

exists. Let $g = f^{-1}$*. Assume that* $0 < c < \infty$*. Then for every* $t \in \mathbb{R}$

$$\lim_{n\to\infty} \frac{1}{\log n} \sum_{i=1}^{n} \frac{1}{i} \mathbb{I}(\mu^*(i)) - g(i) < t) = \int_0^1 F(c(t+z))dz \quad a.s., \tag{3.4}$$

where $F(z) = \exp(-\exp(-z))$*.*

The following result is the a.s. limit theorem for the longest run.

Theorem 3.4.

$$\lim_{n\to\infty} \frac{1}{\log n} \sum_{i=1}^{n} \frac{1}{i} \mathbb{I}(\mu^*(i) - \operatorname{Log} i < t) = \begin{cases} \int_t^{t+1} \exp\left[-\left(\frac{1}{2}\right)^y\right] dy & \text{if } \ p = \frac{1}{2} \\ \int_t^{t+1} \exp\left[-qp^y\right] dy & \text{if } \ p > \frac{1}{2} \end{cases}$$

almost sure.

Proof. We distinguish two cases. First let $p = 1/2$. By Theorem 2.7., $\frac{\tau^*(n)}{2^{n+1}}$ has exponential limit distribution with expectation 1/2, that is $\mathbb{P}(\frac{\tau^*(n)}{2^n} > t) = e^{-t}$.

Now we verify that $\mathbb{E}\mu^*(n) \sim 2^n$. By Remark 3.2, $\lim_{n\to\infty} \mathbb{P}\left(\frac{\tau^*(n)}{E(n)} > t\right) = e^{-t}$. Using the convergence of types theorem (Theorem 2 in Section 10 of Gnedenko-Kolmogorov [10]), we obtain that $\frac{E(n)}{2^n} \to 1$, if $n \to \infty$.

So we can choose in Proposition 3.1 $f(x) = 2^x$, $g(x) = \operatorname{Log} x$. We obtain that $c = \lim_{t\to\infty}(\log f(t))' = \log 2 \in]0, \infty[$. Therefore we can apply Proposition 3.1.

$$\lim_{n\to\infty} \frac{1}{\log n} \sum_{i=1}^{n} \frac{1}{i} \mathbb{I}(\mu^*(i) - \operatorname{Log} i < t) = \lim_{n\to\infty} \frac{1}{\log n} \sum_{i=1}^{n} \frac{1}{i} \mathbb{I}(\mu^*(i) - g(i) < t) =$$

$$\int_0^1 \exp\left[-\exp(-c(t+z))\right] dz = \int_0^1 \exp\left[-\left(\frac{1}{2}\right)^{t+z}\right] dz = \int_t^{t+1} \exp\left[-\left(\frac{1}{2}\right)^{y}\right] dy.$$

Now let $p > 1/2$. By Theorem 2.9,

$$\lim_{n\to\infty} \mathbb{P}(\tau^*(n)qp^n > x) = e^{-x}. \tag{3.5}$$

By Remark 3.2,

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{\tau^*(n)}{E(n)} > x\right) = e^{-x}. \tag{3.6}$$

So $\frac{E(n)}{(qp^n)^{-1}} \to 1$, if $n \to \infty$. Therefore $E(n) \sim (qp^n)^{-1}$. So $f(x) = q^{-1}p^{-x} = \frac{1}{q}\left(\frac{1}{p}\right)^x$. So $g(x) = \operatorname{Log} x + \operatorname{Log} q$ and $c = \log\frac{1}{p}$. This completes the proof of Theorem 3.4. □

References

[1] Berkes, I. and Csáki, E., A universal result in almost sure central limit theory, *Stoch. Proc. Appl.*, 94(1), (2001), 105–134.

[2] Berkes, I., Dehling, H.I. and Móri, T.F., Counterexamles related to the a.s. central limit theorem, *Studia Sci. Math. Hungar.*, 26(1), (1991), 153–164.

[3] Binwanger, K., Embrechts, P., Longest runs in coin tossing, *Mathematics and Economics* 15, (1994), 139–149.

[4] Csáki, E., Földes, A., Komlós, J., Limit theorems for Erdős-Rényi type problems, *Studia Sci. Math. Hungar.*, 22, (1987), 321–332.

[5] Deheuvels, P., On the Erdős-Rényi theorem for random fields and sequences and its relationships with the theory of runs spacings, *Z. Wahrsch. Verw. Gebiete*, 70, no. 1, (1985), 91–115.

[6] Erdős, P., Révész, P., On the lenght of the longest head-run, *Colloquia Mathematica Societatis János Bolyai (16. Topics in information theory, Keszthely (Hungary))*, (1975) 219–228.

[7] Fazekas, I. and Rychlik, Z., Almost sure functional limit theorems, *Ann. Univ. Mariae Curie-Skłodowska, Sect. A*, 56, (2002), 1–18.

[8] Fazekas, I. and Noszály, Cs., Limit theorems for contaminated runs of heads, *(manuscript)* 2007.

[9] Földes, A., The limit distribution of the lenght of the longest head-run, *Periodica Mathematica Hungarica*, 10(4), (1979), 301–310.

[10] Gnedenko, B.V., Kolmogorov, A.N., Limit distributions for sums of independent random variables, *Addison-Wesley Publishing*, London, 1968.

[11] Gordon, L., Schilling, M. F., Waterman, M. S., An extreme value theory for long head runs, *Probability Theory and Related Fields*, 72, (1986), 279–287.

[12] Guibas, L.J. and Odlyzko, A.M., Long repetitive patterns in random sequences, *Z. Wahrsch. Verw. Gebiete*, 53, (1980), 241–262.

[13] Komlós, J., Tusnády, G., On sequences of "pure heads", *The Annals of Probability,* 3, (1975), 273–304.

[14] Kopociński, B., On the distribution of the longest succes-run in Bernoulli trials, *Mat. Stos.*, 34, (1991), 3–13.

[15] Major, P., Almost sure functional limit theorems, Part I. The general case, *Studia Sci. Math. Hungar.*, 34, (1998), 273–304.

[16] Móri, T.F., The a.s. limit distribution of the longest head run, *Can. J. Math.*, 45(6), (1993), 1245–1262.

[17] Móri, T.F., On long run of heads and tails, *Statistics & Probability Letters*, 19, (1994), 85–89.

[18] Móri, T.F., On long run of heads and tails II, *Periodica Mathematica Hungarica*, 28(1), (1994), 79–87.

[19] Muselli, M., Useful inequalities for the longest run distribution, *Statistics & probability letters*, 46, (2000), 239–249.

[20] Philippou, I., Makri, F. S., Successes, Runs and Longest Runs, *Statistics & Probability Letters*, 4, (1986), 211–215.

[21] Schilling, M.F., The longest run of heads, *The College Mathematics Journal*, 21(3), (1990), 196–207.

József Túri
University of Miskolc
Department of Descriptive Geometry
H–3515 Miskolc, Hungary
e-mail: `TuriJ@abrg.uni-miskolc.hu`

Annales Mathematicae et Informaticae
36 *(2009) pp. 143–160*
http://ami.ektf.hu

Investigating the mean response time in finite-source retrial queues using the algorithm by Gaver, Jacobs, and Latouche

Patrick Wüchner[a], János Sztrik[b], Hermann de Meer[a]

[a]Faculty of Informatics and Mathematics, University of Passau, Germany

[b]Faculty of Informatics, University of Debrecen, Hungary

Submitted 18 August 2008; Accepted 1 December 2009

Abstract

In this paper, we discuss the maximum of the mean response time that appears in finite-source retrial queues with orbital search when the arrival rate is varied.

We show that explicit closed-form equations of the mean response time can be derived by exploiting the block-structure of the finite Markov chain underlying the model and using an efficient computational algorithm proposed by Gaver, Jacobs, and Latouche.

However, we also show that already for the discussed relatively simple model, the resulting equation is rather complex which hampers further evaluation.

Keywords: Performance evaluation, Finite-source retrial queues, Closed-form solutions, Orbital search, Block-structured Markov chain, MOSEL-2

1. Introduction

Retrial queues are an important field of study, since in various scenarios, they are able to capture certain behavior of real systems more accurately than classical FCFS queues. Retrial queues are used to model, e.g., telephone traffic in [23], load balancing in multiprotocol label switching (MPLS) networks in [19], Ethernet systems in [2], wireless broadband networks in [20], active queue management of Internet routers in [17], self-organizing peer-to-peer systems in [42], the dynamic host configuration protocol (DHCP) in [24], and mobile communication in [3, 32, 34]. Further application examples are given in [6, 22, 43, 15].

For example, consider a call center scenario with several agents and without a waiting loop installed. If all agents are busy, an additional caller is not able to join a queue, but has to hang up and retry to reach an agent later. Such a retrying caller is said to be in *orbit*.

In addition, consider a call center that is able to log the phone numbers of unserved customers. Then, if an agent gets idle, it may call back unserved orbiting customers. This behavior is called *orbital search*.

In many situations it is unrealistic to assume that the calling population, i.e., the potential number of customers generating requests, is infinitely large. Then, the arrival rate of incoming requests depends on the number of requests already in the system and the arrival process is quasi-random, state-dependent, and non-Poisson. Retrial queues with a finite population size are also known as *finite-source* retrial queues. We are especially interested in models where infinite-source models fail, i.e., models with a small number of sources.

During evaluation of finite-source retrial queues, for some parameter setups a maximum of the mean response time of the system can be identified. Several publications noticed this maximum (e.g., [27, 4, 5, 40]) and gave informal reasons for it (e.g., [40]). Since this maximum should be avoided in real-system configurations by all means, we here try to provide closed-form equations that facilitate the identification of such undesirable configurations during system design.

Our main contribution is the development of novel and explicit closed-form equations for steady-state performance evaluation of the mean response time in finite-source retrial queues with orbital search. To achieve this, we adopt an algorithm introduced by Gaver, Jacobs, and Latouche in [29], which we refer to as *GJL Algorithm*.

The main motivation for this research was to find exact mathematical expressions of the maximum's location in closed form. However, as is discussed later in Section 7, this cannot be achieved, even for the relatively simple model under study.

Previous results on various types of retrial queues are surveyed in [6, 15, 7, 8, 26, 28, 31].

Due to the complexity of finite-source and infinite-source retrial queues, publications on performance measures in closed form are quite rare. Instead, most publications, e.g., the more recent ones [34, 15, 5, 38, 21, 10, 11, 12, 13, 35], employ algorithmic or numerical analysis.

The search for customers immediately on termination of a service was first discussed in the context of classical queues by [33]. More recently, infinite-source retrial queueing systems where the server(s) search for customers after service have been investigated in [21, 16, 25]. We recently introduced and discussed finite-source retrial queues with orbital search by applying numerical analysis in [40] and [41].

There exist several publications discussing infinite-source retrial queues without orbital search and presenting exact results (e.g., [17, 15, 28, 1, 9, 14, 30, 36]), or approximations (e.g., [17, 3, 32, 15, 21]) of performance measures in closed form. Regarding finite-source retrial queues without orbital search, [2] presents closed-form results including phase-type service and multiple servers. However, we are not aware of any publications that present steady-state probabilities and performance measures in closed form applicable to finite-source retrial queues with orbital search.

The remainder of this paper is structured as follows. In Section 2, the investigated model is introduced to fix the notations and preliminary numerical results are presented to state the tackled problem in more detail. Starting from Section 3, we exemplarily focus on the case of three sources and one server. Section 3 discusses the underlying continuous-time Markov chain, and in Section 4, the GJL Algorithm is applied to obtain the steady-state probabilities of the Markov chain in closed form. Based on these equations, in Section 5, mean response time is obtained in closed form and validated in Section 6. In Section 7, we discuss the presented approach with respect to the failure of providing closed-form equations of the maximum's location, and its applicability to derive further performance measures in closed form and for models with a higher number of sources, multiple servers, and phase-type distributed service times. Finally, in Section 8, a conclusion and directions for future work are given.

2. Model description and preliminary numerical analysis

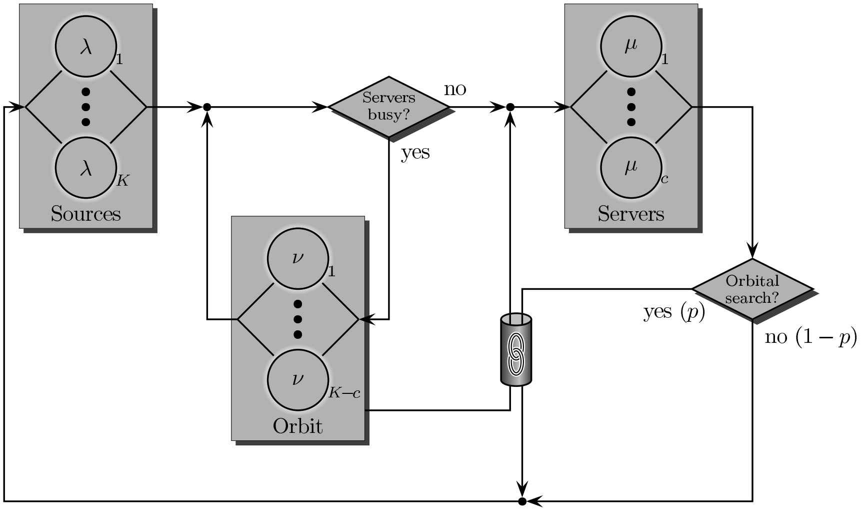


Figure 1: High-level queueing model of finite-source retrial queue with orbital search.

In Fig. 1, a queueing model illustrates the $M/M/c/K/K$ (for Kendall's notation, see [18, p. 242]) finite-source retrial queue with orbital search. All inter-event times involved in the model are assumed to be exponentially distributed. Model extensions by including phase-type distributions are discussed in Section 7.

Each of the K sources is generating primary requests to the retrial queue with rate λ as long as the source is not waiting for a response to an active (i.e., in service or orbiting) request. A primary request first checks whether an idle server is available. If all c identical servers are busy, the primary request enters the orbit instead and retries to get service with rate ν. If a request finds at least one server idle, it starts to receive service with service rate μ. After being serviced, a response is returned to the requesting source. With a probability of p, where $0 \leqslant p \leqslant 1$, at service completion instant, the server carries out orbital search and instantly fetches a request, if available, directly from the orbit (denoted by the link symbol).

The finite-source retrial queue with orbital search can be evaluated numerically quite easily by using the MOSEL-2 performance evaluation tool. The corresponding MOSEL-2 model is shown in Listing 1. The interested reader is referred to [39] for a short introduction to MOSEL-2. In [40, 41] similar models and discussion of MOSEL-2's scalability are presented in the context of finite-source retrial queues.

```
/*** CONSTANTS AND PARAMETERS ************************************************/
CONST     K      := 3;                                  // population size
CONST     mu     := 1;                                  // service rate
PARAMETER lambda := 0.0001, 0.1 .. 1 STEP 0.1;          // request gen. rate
PARAMETER nu     := 0.001, 0.0025, 0.005;               // retrial rate
PARAMETER p      := 1E-8, 0.5, 1-1E-8;                  // search probability

/*** NODES *******************************************************************/
NODE Sources[K]  := K;                                  // the sources
NODE Request[1]  := 0;                                  // primary requests
NODE Server[1]   := 0;                                  // the server
NODE Orbit[K]    := 0;                                  // the orbit
NODE Finished[1] := 0;                                  // response

/*** RULES *******************************************************************/
FROM Sources   TO Request  RATE Sources*lambda;         // primary requests
FROM Request   TO Server   PRIO 1;                      // to server if idle
FROM Request   TO Orbit    PRIO 0;                      // to orbit  if busy
FROM Orbit     TO Server   RATE Orbit*nu;               // retrials
FROM Server    TO Finished RATE mu;                     // service
FROM Finished  TO Sources  WEIGHT 1-p;                  // without orb. search
FROM Finished, Orbit TO Server, Sources WEIGHT p;       // with orbital search

/*** RESULTS *****************************************************************/
PRINT rho        := UTIL(Server);                       // server utilization
PRINT M          := MEAN(Orbit)+MEAN(Server);           // mean # active req.
PRINT S          := K-M;                                // mean # active sources
PRINT N          := MEAN(Orbit);                        // mean # orbit
PRINT ml         := S*lambda;                           // mean throughput
PRINT T          := M/ml;                               // mean response time
PRINT To         := N/ml;                               // mean orbit time
PRINT R          := nu*To;                              // mean # retrials
```

Listing 1: MOSEL-2 model of finite-source retrial queue with orbital search.

Fig. 2 shows the mean response time $\overline{T}$ as a function of request generation rate λ for $K = 3$ sources, $c = 1$ server, service rate $\mu = 1$. We chose different values of retrial rate ν and orbital-search probability p. The curves labeled "num" are obtained by using MOSEL-2's numerical analysis.

These results show a maximum of the mean response time but they are not detailed enough to estimate the exact location of the maximum (in the following denoted as λ_{peak}). This is achieved more accurately by the dashed curves (labeled "expl") which are, in fact, derived using the closed-form equations developed in

Section 5. Hence, in the following, we aim at finding an explicit equation for the mean response time $\overline{T}$ as a function of λ and further model parameters. Afterwards, we discuss whether this equation can be differentiated with respect to λ and whether is is possible to find the roots of the derivation which would lead to an explicit equation of λ_{peak}.

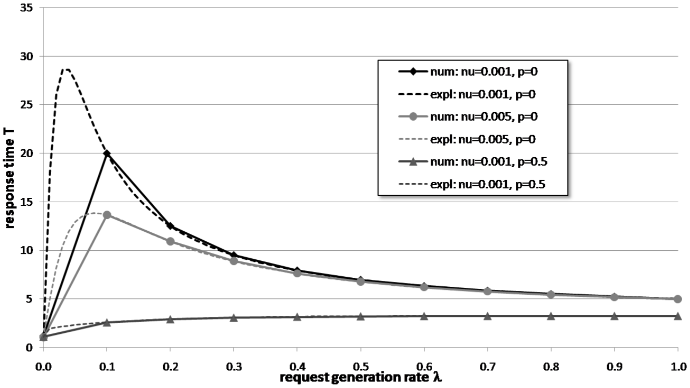


Figure 2: Mean response time $\overline{T}$ over request generation rate λ for service rate $\mu = 1$ and different values of retrial rate ν and orbital-search probability p.

3. Underlying Markov chain

Theorem 3.1. *The behavior of the finite-source retrial queue with orbital search as described in Section 2 can be modeled by a bivariate continuous-time, finite-state Markov chain (CTMC) with state variable $X(t) = (N(t), C(t))$, where variable $N(t)$ is the number of customers in the orbit and variable $C(t)$ is the number of busy servers at time $t \geqslant 0$. Furthermore, this CTMC has a unique steady-state distribution $\pi(i, j)$, with $i = 0, \ldots, K - c$, and $j = 0, \ldots, c$.*

Proof. Due to the memoryless property of the solely exponentially distributed inter-event times, the sojourn times of $X(t)$ are also exponentially distributed, hence the process is a Markov chain.

It is easy to see that $X(t)$ has a finite number of states and is irreducible for all reasonable (i.e., strictly positive) values of λ, μ, and ν. Hence, the underlying stochastic process is positive recurrent which also implies ergodicity. Ergodicity again implies the existence and uniqueness of steady-state probabilities (see [18, p. 69–70]). □

Note that the order of the variables $N(t)$ and $C(t)$ within $X(t)$ is chosen to reflect the structure (levels and phases) of the underlying Markov chain.

In the following, we restrict our investigation to the case $K = 3$ and $c = 1$ to preserve conciseness and traceability. Directions for $K > 3$ and $c > 1$ are given in Section 7. The state transition diagram of the corresponding CTMC is shown in Fig. 3.

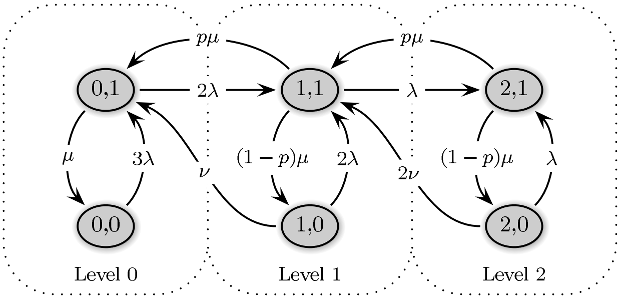


Figure 3: State transition diagram of finite-source retrial queue with orbital search for $K = 3$ and $c = 1$.

Note that for $p = 0$, Fig. 3 reduces to the state transition diagram of the classical $M/M/1/3/3$ finite-source retrial queue without orbital search. On the other hand, for $p = 1$, Fig. 3 reduces to the state transition diagram of the classical $M/M/1/3/3$–First-Come-First-Served (FCFS) queue.

The CTMC shown in Fig. 3 can be structured according to levels reflecting the number of customers in the orbit $N(t)$. Each level consists of two phases indicating the state of the server given by $C(t)$. Moreover, the CTMC constitutes a finite quasi-birth-death process (QBD), which is skip-free in both directions. This structure is also reflected in the block-tridiagonal form of the infinitesimal generator matrix $\mathbf{Q}$ of the CTMC given by

$$\mathbf{Q} = \begin{pmatrix} \mathbf{A}^{(0)} & \mathbf{\Lambda}^{(0)} & \mathbf{0} \\ \mathbf{M}^{(1)} & \mathbf{A}^{(1)} & \mathbf{\Lambda}^{(1)} \\ \mathbf{0} & \mathbf{M}^{(2)} & \mathbf{A}^{(2)} \end{pmatrix}, \tag{3.1}$$

where the sub-matrices

$$\begin{aligned}
\mathbf{A}^{(0)} &= \begin{pmatrix} -3\lambda & 3\lambda \\ \mu & -2\lambda-\mu \end{pmatrix}, & \mathbf{\Lambda}^{(0)} &= \begin{pmatrix} 0 & 0 \\ 0 & 2\lambda \end{pmatrix}, \\
\mathbf{A}^{(1)} &= \begin{pmatrix} -2\lambda-\nu & 2\lambda \\ (1-p)\mu & -\lambda-\mu \end{pmatrix}, & \mathbf{\Lambda}^{(1)} &= \begin{pmatrix} 0 & 0 \\ 0 & \lambda \end{pmatrix}, \\
\mathbf{A}^{(2)} &= \begin{pmatrix} -\lambda-2\nu & \lambda \\ (1-p)\mu & -\mu \end{pmatrix}, & \mathbf{M}^{(1)} &= \begin{pmatrix} 0 & \nu \\ 0 & p\nu \end{pmatrix}, \\
\mathbf{0} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, & \mathbf{M}^{(2)} &= \begin{pmatrix} 0 & 2\nu \\ 0 & p\nu \end{pmatrix},
\end{aligned}$$

can be obtained by inspecting the transition rates given in Fig. 3. Note that the notation of the sub-matrices is chosen in accordance to [29].

4. Application of the GJL algorithm

To obtain the steady-state probabilities in closed form, we apply the computational algorithm proposed in [29]. For this, we exploit the relatively simple structure of the underlying Markov chain as presented in Section 3. For a thorough explanation and proof of the GJL algorithm, we refer the interested reader to [29].

The GJL Algorithm is applied to the structured CTMC (i.e., finite QBD) given in Sect. 3, where $K = 3$:

1. Calculation of $\mathbf{C}_n$ with $0 \leqslant n \leqslant 2$:

$$\mathbf{C}_0 = \mathbf{A}^{(0)} = \begin{pmatrix} -3\lambda & 3\lambda \\ \mu & -2\lambda - \mu \end{pmatrix}, \tag{4.1}$$

$$\mathbf{C}_1 = \mathbf{A}^{(1)} + \mathbf{M}^{(1)} \left(-\mathbf{C}_0^{-1} \boldsymbol{\Lambda}^{(0)} \right)$$
$$= \begin{pmatrix} -2\lambda - \nu & 2\lambda + \nu \\ (1-p)\mu & -\lambda - (1-p)\mu \end{pmatrix}, \tag{4.2}$$

$$\mathbf{C}_2 = \mathbf{A}^{(2)} + \mathbf{M}^{(2)} \left(-\mathbf{C}_1^{-1} \boldsymbol{\Lambda}^{(1)} \right)$$
$$= \begin{pmatrix} -\lambda - 2\nu & \lambda + 2\nu \\ (1-p)\mu & -(1-p)\mu \end{pmatrix}. \tag{4.3}$$

2. Obtaining $\boldsymbol{\pi}_2$: Since the system $\boldsymbol{\pi}_2 \mathbf{C}_2 = (0, 0)$ is linearly dependent, we can replace one equation of the system by the normalization condition $\boldsymbol{\pi}_2 \binom{1}{1} = 1$ and solve

$$\boldsymbol{\pi}_2 \begin{pmatrix} -\lambda - 2\nu & 1 \\ (1-p)\mu & 1 \end{pmatrix} = (0, 1), \tag{4.4}$$

instead. This leads to

$$\boldsymbol{\pi}_2 = \tfrac{1}{\lambda + (1-p)\mu + 2\nu} \left((1-p)\mu \;\; \lambda + 2\nu \right). \tag{4.5}$$

3. Obtaining $\mathbf{P}_n$, $n = 2, 1, 0$, recursively:

$$\mathbf{P}_2 = \boldsymbol{\pi}_2$$
$$= \tfrac{1}{\lambda + (1-p)\mu + 2\nu} \left((1-p)\mu \;\; \lambda + 2\nu \right), \tag{4.6}$$

$$\mathbf{P}_1 = \mathbf{P}_2 \mathbf{M}^{(2)} \left(-\mathbf{C}_1^{-1} \right)$$
$$= \tfrac{1}{\lambda + (1-p)\mu + 2\nu}$$
$$\cdot \left(\tfrac{(1-p)\mu^2 (\lambda p + 2\nu)}{\lambda (2\lambda + \nu)} \;\; \tfrac{\mu (\lambda p + 2\nu)}{\lambda} \right), \tag{4.7}$$

$$\mathbf{P}_0 = \mathbf{P}_1 \mathbf{M}^{(1)} \left(-\mathbf{C}_0^{-1} \right)$$
$$= \tfrac{1}{\lambda + (1-p)\mu + 2\nu}$$
$$\cdot \left(\tfrac{\mu^3 (\lambda p + 2\nu)(2\lambda p + \nu)}{6\lambda^3 (2\lambda + \nu)} \;\; \tfrac{\mu^2 (2\nu + \lambda p)(2\lambda p + \nu)}{2\lambda^2 (2\lambda + \nu)} \right). \tag{4.8}$$

4. Re-normalizing vector $\mathbf{P} = (\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2)$: For this, we derive the normalization constant P_N as follows:

$$\begin{aligned} P_N &= \mathbf{P}_0 \tbinom{1}{1} + \mathbf{P}_1 \tbinom{1}{1} + \mathbf{P}_2 \tbinom{1}{1} \\ &= \tfrac{1}{\lambda+(1-p)\mu+2\nu} \tfrac{1}{6\lambda^3(2\lambda+\nu)} \widetilde{P_N}, \end{aligned} \tag{4.9}$$

where $\widetilde{P_N}$ is given by the term

$$\begin{aligned} \widetilde{P_N} = {} & 2\mu^3\nu^2 + 5\mu^3\nu\lambda p + 2\mu^3\lambda^2 p^2 + 6\mu^2\lambda\nu^2 \\ & + 3\mu^2\lambda^2\nu p + 12\mu^2\lambda^2\nu + 6\mu^2\lambda^3 p\nu \\ & + 30\mu\lambda^3 + 12\mu\lambda^2\nu^2 + 12\mu\lambda^4 + 12\lambda^5 \\ & + 30\lambda^4\nu + 12\lambda^3\nu^2. \end{aligned} \tag{4.10}$$

In the following, we denote by $P_{i,j}, i \in \{0,1,2\}, j \in \{0,1\}$, the j-th element of vector $\mathbf{P}_i$ and with $\pi(i,j)$ the steady-state probability of state (i,j), i.e., phase j in level i. With $\mathbf{P}_i$ given by Eqs. (4.6) through (4.8) and P_N given by Eq. (4.9), the desired steady-state probabilities of the CTMC depicted in Fig. 3 can be derived in closed form as follows:

$$\pi(0,0) = \frac{P_{0,0}}{P_N} = \frac{\mu^3(\lambda p + 2\nu)(2\lambda p + \nu)}{\widetilde{P_N}}, \tag{4.11}$$

$$\pi(0,1) = \frac{P_{0,1}}{P_N} = \frac{3\lambda\mu^2(\lambda p + 2\nu)(2\lambda p + \nu)}{\widetilde{P_N}}, \tag{4.12}$$

$$\pi(1,0) = \frac{P_{1,0}}{P_N} = \frac{6\lambda^2(1-p)\mu^2(\lambda p + 2\nu)}{\widetilde{P_N}}, \tag{4.13}$$

$$\pi(1,1) = \frac{P_{1,1}}{P_N} = \frac{6\lambda^2\mu(\lambda p + 2\nu)(2\lambda + \nu)}{\widetilde{P_N}}, \tag{4.14}$$

$$\pi(2,0) = \frac{P_{2,0}}{P_N} = \frac{6\lambda^3(1-p)\mu(2\lambda + \nu)}{\widetilde{P_N}}, \tag{4.15}$$

$$\pi(2,1) = \frac{P_{2,1}}{P_N} = \frac{6\lambda^3(\lambda + 2\nu)(2\lambda + \nu)}{\widetilde{P_N}}. \tag{4.16}$$

5. Mean response time in closed form

In Section 4, closed-form expressions of the steady-state probabilities of the underlying Markov chain were derived. These expressions are now used to obtain the mean response time $\overline{T}$ in closed form.

Mean number of active requests $\overline{M}$: The mean number of requests located in service or in orbit is given by

$$\overline{M} = \pi(0,1) + \pi(1,0) + 2\pi(1,1) + 2\pi(2,0)$$

$$
\begin{aligned}
&+ 3\pi(2,1)\\
&= \frac{3\lambda}{\widetilde{P_N}}(\mu^2\lambda p\nu + 2\mu^2\nu^2 + 2\lambda^2\mu^2 p + 4\lambda\mu^2\nu\\
&\quad + 20\lambda^2\mu\nu + 8\lambda\mu\nu^2 + 8\lambda^3\mu + 12\lambda^4\\
&\quad + 30\lambda^3\nu + 12\lambda^2\nu^2). \qquad (5.1)
\end{aligned}
$$

Mean system throughput $\overline{\lambda}$: The mean throughput of the finite-source retrial queue with orbital search can be obtained from

$$
\begin{aligned}
\overline{\lambda} &= (K - \overline{M})\lambda\\
&= 3\lambda\Bigg(1 - \frac{\lambda}{\widetilde{P_N}}(\mu^2\lambda p\nu + 2\mu^2\nu^2 + 2\lambda^2\mu^2 p\\
&\quad + 4\lambda\mu^2\nu + 20\lambda^2\mu\nu + 8\lambda\mu\nu^2 + 8\lambda^3\mu\\
&\quad + 12\lambda^4 + 30\lambda^3\nu + 12\lambda^2\nu^2)\Bigg)\\
&= \frac{3\lambda\widetilde{\Lambda}}{\widetilde{P_N}}, \qquad (5.2)
\end{aligned}
$$

where $\widetilde{\Lambda}$ is defined as follows:

$$
\begin{aligned}
\widetilde{\Lambda} &= \mu(2\mu^2\lambda^2p^2 + 5\mu^2\lambda p\nu + 2\mu^2\nu^2 + 4\lambda^3\mu p\\
&\quad + 2\lambda^2\mu p\nu + 8\lambda^2\mu\nu + 4\lambda\mu\nu^2 + 4\lambda^4\\
&\quad + 10\lambda^3\nu + 4\lambda^2\nu^2). \qquad (5.3)
\end{aligned}
$$

Mean response time $\overline{T}$: The mean time spent by each request in the orbit and the server can also be calculated by applying Little's Law as follows:

$$
\begin{aligned}
\overline{T} &= \frac{\overline{M}}{\overline{\lambda}}\\
&= \frac{1}{\widetilde{\Lambda}}(\mu^2\lambda p\nu + 2\mu^2\nu^2 + 2\lambda^2\mu^2 p + 4\lambda\mu^2\nu\\
&\quad + 12\lambda^4 + 20\lambda^2\mu\nu + 8\lambda\mu\nu^2 + 8\lambda^3\mu\\
&\quad + 30\lambda^3\nu + 12\lambda^2\nu^2) \qquad (5.4)
\end{aligned}
$$

In Fig. 4, we exemplarily plot the mean response time $\overline{T}$ as a function of request generation rate λ and retrial rate ν for service rate $\mu = 1$ and orbital-search probability $p = 0.5$ by employing Eq. (5.4). The presented closed-form equations facilitate the retrieval of fine-grained results since, in general, they can be implemented more efficiently than numerical analysis.

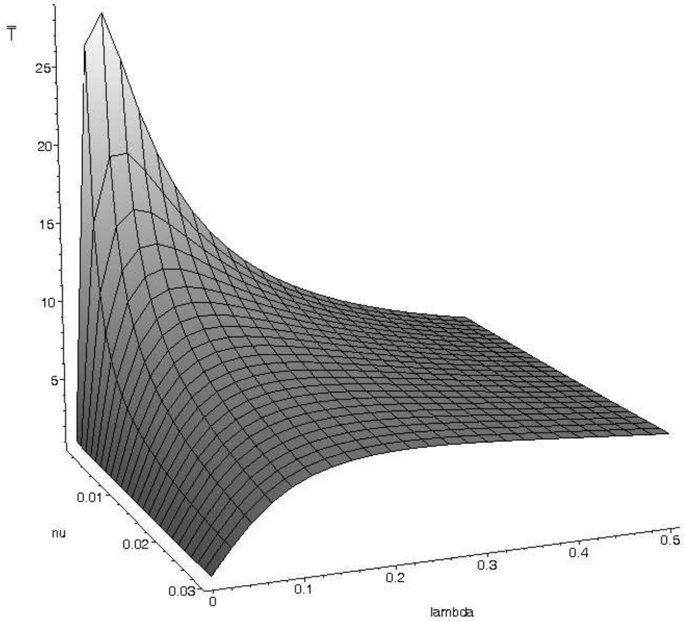


Figure 4: Mean response time $\overline{T}$ (z-axis) over request generation rate λ (x-axis) and retrial rate ν (y-axis) for service rate $\mu = 1$ and orbital-search probability $p = 0.5$.

6. Validation of closed-form equations

In this section, the closed-form equations derived in Section 5 are validated against numerical results and against well-known closed-form equations of $M/M/1/K/K$–$FCFS$ queueing systems.

6.1. Comparison to numerical results

Table 1 compares results obtained from numerical analysis using MOSEL-2 (see Section 2) to results obtained by using the closed-form expressions presented in Section 5 for $\lambda = 0.1$, $\nu = 0.0025$, $\mu = 1$, and $p = 0.5$. It can be seen that the numerical results are very close to the closed-form results.

6.2. Comparison to $M/M/1/K/K$–$FCFS$ system

As already mentioned in Section 3, the state-transition diagram given in Fig. 3 takes the form of the CTMC underlying an $M/M/1/3/3$–$FCFS$ queueing system for $p = 1$. Such finite-population FCFS systems are also known as *Machine Repairman Models* (see [18, p. 252]), for which performance measures are available in

Perf. Measure	Num. Analysis	Closed-Form Expression
ρ	0.239555	0.2395547133
$\overline{M}$	0.604453	0.6044528670
$\overline{S}$	2.39555	2.395547133
$\overline{N}$	0.364898	0.3648981538
$\overline{\lambda}$	0.239555	0.2395547133
$\overline{T}$	2.52324	2.523235126
$\overline{T}_O$	1.52324	1.523235126
$\overline{R}$	0.00380809	0.003808087814

Table 1: Model results for $\lambda = 0.1$, $\nu = 0.0025$, $\mu = 1$, and $p = 0.5$.

closed form.

According to [18], the mean response time $\overline{T}$ of an $M/M/1/K/K$–$FCFS$ queue is given by

$$\overline{T}_{\text{FCFS}} = \frac{K}{\mu(1-\pi_0)} - \frac{1}{\lambda}, \tag{6.1}$$

where the steady-state probability of an idle server π_0 is given by

$$\pi_0 = \frac{1}{\sum\limits_{k=0}^{K} \left(\frac{\lambda}{\mu}\right)^k \frac{K!}{(K-k)!}}. \tag{6.2}$$

In the current scenario, where $K = 3$, Eq. (6.1) can be rewritten as

$$\begin{aligned} \overline{T}_{\text{FCFS}} &= \frac{3}{\mu(1-\pi_0)} - \frac{1}{\lambda} \\ &= \frac{6\lambda^2 + 4\mu\lambda + \mu^2}{(\lambda^2 + 2\mu\lambda + \mu^2)\mu}. \end{aligned} \tag{6.3}$$

When setting $p = 1$ in Eq. (5.4), we equivalently get

$$\begin{aligned} \overline{T} = {} & \Big(5\mu^2\lambda\nu + 2\mu^2\nu^2 + 2\mu^2\lambda^2 + 20\lambda^2\mu\nu \\ & + 8\lambda\mu\nu^2 + 8\mu\lambda^3 + 12\lambda^4 + 30\lambda^3\nu + 12\lambda^2\nu^2\Big) \\ & \Big/ \Big(\mu\big(5\mu^2\lambda\nu + 2\mu^2\lambda^2 + 2\mu^2\nu^2 + 4\mu\lambda^3 \\ & + 10\lambda^2\mu\nu + 4\lambda\mu\nu^2 + 4\lambda^4 + 10\lambda^3\nu \\ & + 4\lambda^2\nu^2\big)\Big) \\ = {} & \frac{6\lambda^2 + 4\mu\lambda + \mu^2}{(\lambda^2 + 2\mu\lambda + \mu^2)\mu} \\ = {} & \overline{T}_{\text{FCFS}}. \end{aligned} \tag{6.4}$$

Hence, the two closed forms match in the case of $p = 1$. For the sake of completeness, we compare the mean response time $\overline{T}_p$ of an $M/M/1/3/3$ retrial queue with orbital search ($\mu = 1$, $\nu = 0.001$, $p = 0.1 \dots 0.9$) with the mean response time $\overline{T}_{\mathrm{FCFS}}$ of an $M/M/1/3/3$–$FCFS$ queue ($\mu = 1$) in Fig. 5.

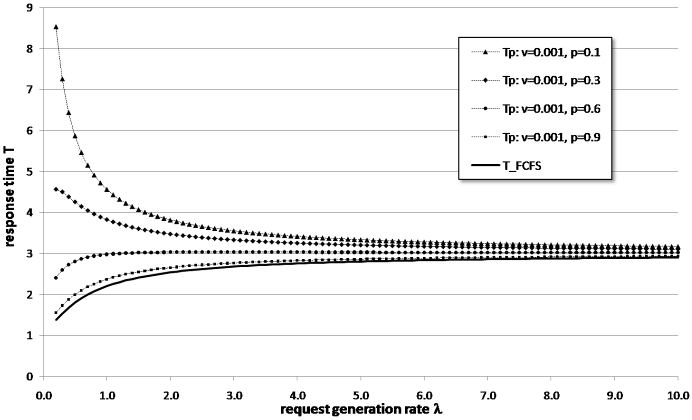


Figure 5: Mean response times $\overline{T}_p$ and $\overline{T}_{\mathrm{FCFS}}$ over request generation rate λ.

As expected, $\overline{T}_p$ gets close to $\overline{T}_{\mathrm{FCFS}}$ for $p \approx 1$. It can also be seen that all curves get close to each other for high values of the request generation rate λ. High generation rates lead to high server utilization. The server is then kept busy by primary requests even if the orbital search probability p is low. Also for high values of ν (compared to μ), the behavior of finite-source retrial queues with orbital search should be close to the behavior of an $M/M/1/K/K$–$FCFS$ queueing system. This statement is confirmed by Fig. 6, where the mean response times $\overline{T}_\nu$ of an $M/M/1/3/3$ retrial queue with orbital search ($\mu = 1$, $\nu = 0.1 \dots 20$, $p = 0.1$), and $\overline{T}_{\mathrm{FCFS}}$ of the $M/M/1/3/3$–$FCFS$ queue ($\mu = 1$) are compared.

It can be seen that for high values of ν, $\overline{T}_\nu$ gets close to $\overline{T}_{\mathrm{FCFS}}$. Again, for high server utilization, $\overline{T}_\nu$ becomes independent of ν.

Note that all results of Figs. 5 and 6 are obtained using Eqs. (5.4) and (6.3).

7. Discussion of approach

7.1. Location of maximum

To find an equation for λ_{peak}, i.e., the arrival rate of the maximum mean response time, in closed form, we need to find the roots of equation $\frac{d\overline{T}}{d\lambda}$. However, according to Eq. (5.4), $\overline{T}$ is quite complex already for this simple model. The resulting equation derived from $\frac{d\overline{T}}{d\lambda}$ is a ratio of high order polynomials for which the roots could be found numerically, but unfortunately not in a closed form.

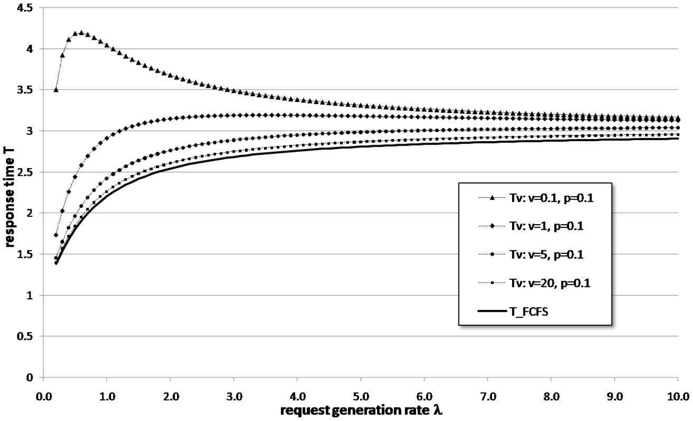


Figure 6: Mean response times $\overline{T}_\nu$ and $\overline{T}_{\text{FCFS}}$ over request generation rate λ.

7.2. Further performance measures

Unfortunately, our forseen goal to provide a closed-form equation for the maximum's location cannot be achieved. However, by using the steady-state probabilities presented in Section 4, further performance measures of the discussed retrial queue can be derived in closed form.

For example, Eqs. (4.11) through (4.16) can be readily used together with the equations provided in our previous work [40, Sec. 2.3] to obtain steady-state performance measures like the server utilization, the mean number of orbiting customers, the mean waiting time, etc. in closed form.

7.3. Model generalization

While in Sections 3 through 6, for the sake of clearness, the investigation is restricted to $K = 3$ and $c = 1$ to show the principles, we now discuss the applicability of the method for a higher number of sources and servers as well as phase-type service.

7.3.1. Increasing the number of sources

The GJL Algorithm employed in Section 4 can be applied in principle also for higher values of K. If K is increased, the number of levels of the underlying CTMC (recall Fig. 3) increases, but the number of phases in each level stays the same, i.e., two. As a consequence, matrix $\mathbf{Q}$ (recall Eq. (3.1)) will grow by one additional column and one additional row of 2×2 sub-matrices per each additional source. The number of the matrices $\mathbf{C}_n$ increases ($0 \leqslant n \leqslant K-1$) but not their size ($2 \times 2$). This results in additional iteration steps in Steps 1 and 3 of the GJL Algorithm but the matrices $\mathbf{C}_n$ can still be inverted explicitly in a relatively compact way.

7.3.2. Increasing the number of servers

If the number of servers is increased, then also the size of square matrices $\mathbf{C}_n$ increases. By using, e.g., Eq. (7.1) (cf. [37]):

$$\mathbf{C}_n^{-1} = \frac{1}{det(\mathbf{C}_n)} adj(\mathbf{C}_n), \tag{7.1}$$

the matrices $\mathbf{C}_n$ can still be inverted explicitly. This, however, increases the effort and leads to even more complex closed-form equations.

7.3.3. Increasing the number of service phases

The method can also be used in case of a single server which conducts phase-type service with a finite number of service phases. Comparable to Section 7.3.2, this results in additional phases within the Markov chain and in larger $\mathbf{C}_n$ matrices, which can still be inverted explicitly. The proposed method cannot be applied directly to multiple-server retrial queues with phase-type service, since this implies higher-dimensional Markov chains.

8. Conclusion and future work

In this paper, we present steady-state probabilities and the mean response time of single-server finite-source retrial queues with orbital search and three sources in closed form. The equations are derived by adopting an algorithm introduced in [29]. The results are validated against results obtained by numerical analysis and against closed-form equations well-known for $M/M/1/K/K\text{–}FCFS$ queueing systems.

It could be shown that due to the high complexity of the derived equations, it is not possible to derive the location of the mean response time's maximum in closed form. However, using the derived closed-form equations of the steady-state probabilities gives raise to other interesting performance measures in closed-form as well.

Our planned future work includes applying the algorithm to a higher number of sources and servers, phase-type service, and unreliable servers. It may also be worthwhile to study approximate solutions for higher numbers of sources, servers, and service phases.

9. Acknowledgments

This research is partially supported by the German-Hungarian Intergovernmental Scientific Cooperation, HAS-DFG, 436 UNG 113/197/0-1, by the Hungarian Scientific Research Fund, OTKA K60698/2006, by the AutoI project (STREP, FP7 Call 1, ICT-2007-1-216404), by the ResumeNet project (STREP, FP7 Call 2, ICT-2007-2-224619), by the SOCIONICAL project (IP, FP7 Call 3, ICT-2007-3-231288), and by the EuroNF Network of Excellence (FP7, IST 216366).

The first author is especially grateful to the participants of the Dagstuhl Seminar on "Numerical Methods for Structured Markov Chains" (07461) for very fruitful discussions.

References

[1] Aissani, A., Artalejo, J., "On the single server retrial queue subject to breakdowns," *Queueing Systems*, vol. 30, (1998) 309–321.

[2] Alfa, A.S., Isotupa, K.S., "An M/PH/k retrial queue with finite number of sources," *Computers and Operations Research*, vol. 31, (2004) 1455–1464.

[3] Alfa, A.S., Li, W., "PCS networks with correlated arrival process and retrial phenomenon," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, (October 2002) 630–637.

[4] Almasi, B., Bolch, G., Sztrik, J., "Heterogeneous finite-source retrial queues," University of Erlangen-Nuremberg, Erlangen, Germany, Tech. Rep. TR-I4-02-04, 2004.

[5] Almasi, B., Roszik, J., Sztrik, J., "Homogeneous finite-source retrial queues with server subject to breakdowns and repairs," *Mathematical and Computer Modelling*, vol. 42, (2005) 673–682.

[6] Artalejo, J.R., "Retrial queues with a finite number of sources," *J. Korean Math. Soc.*, vol. 35, (1998) 503–525.

[7] Artalejo, J.R., "Accessible bibliography on retrial queues," *Mathematical and Computer Modelling*, vol. 30, (1999) 1–6.

[8] Artalejo, J.R., "A classified bibliography of research on retrial queues: Progress in 1990-1999," *TOP*, vol. 7, (1999) 187–211.

[9] Artalejo, J.R., "Stationary analysis of the characteristics of the M/M/2 queue with constant repeated attempts," *Opsearch*, vol. 33, no. 2, (1996) 83–95.

[10] Artalejo, J.R., Chakravarthy, S.R., "Computational analysis of the maximal queue length in the MAP/M/c retrial queue," *Applied Mathematics and Computation*, vol. 183, (2006) 1399–1409.

[11] Artalejo, J.R., Chakravarthy, S.R., "Algorithmic analysis of the maximum level length in general-block two-dimensional Markov processes," *Mathematical Problems in Engineering*, vol. 2006, (2006) 1–15.

[12] Artalejo, J.R., Chakravarthy, S.R., "Algorithmic analysis of the MAP/PH/1 retrial queue," *TOP*, vol. 14, no. 2, (2006) 293–332.

[13] Artalejo, J.R., Chakravarthy, S.R., Lopez-Herrero, M.J., "The busy period and the waiting time analysis of a MAP/M/c queue with finite retrial group," *Stochastic Analysis and Applications*, vol. 25, (2007) 445–469.

[14] Artalejo, J.R., Economou, A., Lopez-Herrero, M.J., "Algorithmic analysis of the maximum queue length in a busy period for the M/M/c retrial queue," *INFORMS J. on Computing*, vol. 19, no. 1, (2007) 121–126.

[15] Artalejo, J.R., Gómez-Corral, A., *Retrial Queueing Systems: A Computational Approach.* Springer Verlag, 2008.

[16] Artalejo, J.R., Joshua, V.C., Krishnamoorthy, A., "An M/G/1 retrial queue with orbital search by the server," in *Advances in Stochastic Modelling*, J. R. Artalejo and A. Krishnamoorthy, Eds. NJ: Notable Publications Inc., (2002) 41–54.

[17] Avrachenkov, K., Yechiali, U., "Retrial networks with finite buffers and their application to internet data traffic," *Probability in the Engineering and Informational Sciences*, vol. 22, (2008) 519–536.

[18] Bolch, G., Greiner, S., Meer, H., Trivedi, K., *Queueing Networks and Markov Chains*, 2nd ed. New York: John Wiley & Sons, 2006.

[19] Chakka, R., Do, T.V., "The $MM\sum_{k=1}^{K} CPP_k/GE/c/L$ G-Queue and Its Application to the Analysis of the Load Balancing in MPLS Networks." in *Proc. of 27th Annual IEEE Conference on Local Computer Networks (LCN 2002), 6-8 November 2002*, Tampa, FL, USA, (2002) 735–736.

[20] Chakka, R., Do, T.V., "The MM $\sum_{k=1}^{K} CPP_k/GE/c/L$ G-queue with heterogeneous servers: Steady state solution and an application to performance evaluation," *Performance Evaluation*, vol. 64, (March 2007) 191–209.

[21] Chakravarthy, S.R., Krishnamoorthy, A., Joshua, V., "Analysis of a multiserver retrial queue with search of customers from the orbit," *Performance Evaluation*, vol. 63, no. 8, (2006) 776–798.

[22] Choi, B.D., Chang, Y., "Single server retrial queues with priority calls," *Mathematical and Computer Modelling*, vol. 30, (1999, invited paper) 7–32.

[23] Cohen, J.W., "Basic problems of telephone traffic theory and the influence of repeated calls," *Philips Telecommun. Rev.*, vol. 18, no. 2, (1957) 49–100.

[24] Do, T.V., "An efficient solution to a retrial queue for the performability evaluation of DHCP," *Computers and Operations Research*, vol. In Press, Corrected Proof, (2009) [Online]. Available: `http://www.sciencedirect.com/science/article/B6VC5-4WGVPXN-1/2/48b8e2e8550847fd28abec83ff41f72d`

[25] Dudin, A.N., Krishnamoorthy, A., Joshua, V., Tsarenkov, G.V., "Analysis of the BMAP/G/1 retrial system with search of customers from the orbit." *Eur. J. Operational Research*, vol. 157, no. 1, (2004) 169–179.

[26] Falin, G., "A survey of retrial queues," *Queueing Systems*, vol. 7, no. 2, (1990) 127–167.

[27] Falin, G., Artalejo, J., "A finite source retrial queue," *Eur. J. Operational Research*, vol. 108, (1998) 409–424.

[28] Falin, G., Templeton, J., *Retrial Queues.* Chapman & Hall, 1997.

[29] Gaver, D., Jacobs, P., Latouche, G., "Finite birth-and-death models in randomly changing environments," *Adv. Appl. Prob.*, vol. 16, (1984) 715–731.

[30] Hanschke, T., "Explicit formulas for the characteristics of the M/M/2/2 queue with repeated attempts," *Appl. Prob.*, vol. 24, (1987) 486–494.

[31] Kulkarni, V.G., Liang, H.M., *Frontiers in Queueing: Models and Applications in Science and Engineering.* CRC Press, 1997, ch. Retrial queues revisited, 19–34.

[32] Marsan, M.A., Carolis, G., Leonardi, E., Lo Cigno, R., Meo, M., "Efficient estimation of call blocking probabilities in cellular mobile telephony networks with customer retrials," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 2, (February 2001) 332–346.

[33] Neuts, M.F., Ramalhoto, M.F., "A service model in which the server is required to search for customers," *Appl. Prob.*, vol. 21, no. 1, (March 1984) 157–166.

[34] Roszik, J., Kim, C., Sztrik, J., "Retrial queues in the performance modeling of cellular mobile networks using MOSEL," *International Journal of Simulation: Systems, Science and Technology*, vol. 6, (2005) 38–47.

[35] Roszik, J., Sztrik, J., Virtamo, J., "Performance analysis of finite-source retrial queues operating in random environments," *Int. J. Operational Research*, vol. 2, no. 3, (2007) 254–268.

[36] Sherman, N.P., Kharoufeh, J.P., "An M/M/1 retrial queue with unreliable server," *Operations Research Letters*, vol. 34, (2006) 697–705.

[37] Stewart, G.W., "On the adjugate matrix," *Linear Algebra and its Applications*, vol. 283, (1998) 151–164.

[38] Sztrik, J., Almasi, B., Roszik, J., "Heterogeneous finite-source retrial queues with server subject to breakdowns and repairs," *Math. Sciences*, vol. 132, (2006) 677–685.

[39] Wüchner, P., Meer, H., Barner, J., Bolch, G., "A brief introduction to MOSEL-2," in *Proc. of MMB 2006 Conference*, R. German and A. Heindl, Eds., GI/ITG/MMB, University of Erlangen. VDE Verlag, 2006.

[40] Wüchner, P., Sztrik, J., Meer, H., "Homogeneous finite-source retrial queues with search of customers from the orbit," in *Proc. of 14th GI/ITG Conference on Measurement, Modelling and Evaluation of Computer and Communication Systems (MMB 2008)*, Dortmund, Germany, March 2008.

[41] Wüchner, P., Sztrik, J., Meer, H., "Finite-source M/M/S retrial queue with search for balking and impatient customers from the orbit," *Computer Networks*, vol. 53, (2009) 1264–1273.

[42] Wüchner, P., Sztrik, J., Meer, H., "The impact of retrials on the performance of self-organizing systems," *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, vol. 31, no. 1, (March 2008) 29–33.

[43] Yang, T., Templeton, J.G.C., "A survey on retrial queues," *Queueing Systems*, vol. 2, (1987) 201–233.

Patrick Wüchner, Hermann de Meer
Faculty of Informatics and Mathematics, University of Passau
Innstraße 43, 94032 Passau, Germany
e-mail: `{patrick.wuechner,hermann.demeer}@uni-passau.de`

János Sztrik
Faculty of Informatics, University of Debrecen
Egyetem tér 1, P.O. Box 12
4010 Debrecen, Hungary
e-mail: `jsztrik@inf.unideb.hu`

Methodological papers

Annales Mathematicae et Informaticae
36 *(2009) pp. 163–174*
http://ami.ektf.hu

Teaching programming language in grammar schools

Zoltán Hernyák, Roland Király

Eszterházy Károly College, Department of Information Technology

Submitted 6 October 2009; Accepted 10 December 2009

Abstract

In Hungary algorithmical thinking is a part of teaching informatics both in primary and secondary grammar schools. A teacher usually starts with some everyday algorithm, taking examples from cooking or solving a mathematical or physical problem. The steps of the solutions are usually represented in a flow diagram. This diagram is a graphical representation of the algorithm steps including decision symbols. With these decision symbols, selections and iterations can be applied.

Unfortunately, the common ways of describing algorithms are far from functional thinking, therefore it is rather difficult for teachers to find materials on teaching functional programming. On the other hand, programming and trying the algorithm in a functional way is much easier as in the imperative way ([3], [4]).

The next step is usually the description of the algorithm by a sentence-like language, which is very close to BASIC programming language. At this point the teacher switches to a programming language, like Pascal, BASIC, C# [17], or any OOP [9] supportive or OOP language [9], all of which are imperative languages.

These languages were taught to teachers during their studies, and are used in their workplaces, in grammar schools as well. We believe that the functional programming paradigm is raising nowadays, and getting more and more important. In this paper we are trying to show and prove that this programming style is appropriate for teaching programming in grammar schools.

1. Introduction

The mathematical fundamentals of functional programming are based on Church's lambda-calculus theory which he invented in 1932–33. Turing proved that the effectively evaluable functions interpreted on non-negative integers following the lambda-calculus are the same as the ones that are computable by the Turing machine commonly utilized by imperative languages. Thus every task which can be solved by imperative languages are as well solvable in functional languages, and vice versa.

There are several functional languages, which can be used to code the algorithms. Most of them can be downloaded and use for free, and has some sort of IDE (Integrated Development Environment).

Erlang is a development of Ericson and Ellemtel Computer Science Laboratories. Erlang is a programming language in that it is possible to develop concurrent, real-time, distributed and highly error-tolerant systems. Ericson uses the Open Telecom Platform extension of Erlang to develop telecommunication systems. The language has internal methods to achieve that without shared memory distributed applications communicate through signaling among themselves. It supports integrating components written in different programming languages, but is generally a weakly typed language.

Haskell is an advanced purely functional programming language. An open source product of more than twenty years of cutting edge research, it allows rapid development of robust, concise, correct software. With strong support for integration with other languages, built-in concurrency and parallelism, debuggers, profilers, rich libraries and an active community, Haskell makes it easier to produce flexible, maintainable high-quality software.

Clean is a non-profit development as a functional programming language, with many similarities to Haskell. With the ObjectIO library extension of Clean, one can develop interactive applications having menus, and dialog windows.

In Hungary at Eötvös Lóránd University the functional programming paradigm is used both in education and in scientific researches and projects. There are attempts to include functional languages in education abroad too.

We use Clean functional programming language as a reference language to solve some basic algorithmic problems. The Clean System is a software development system for developing applications in Clean. The Clean System is available on many platforms (PC, Mac, SUN) and operating systems (Windows'95/'98/2000/NT, Linux, MacOS, Solaris). The main platforms are PC and Mac. The Clean System is a full-fledged system that can be used in industrial environments. The Clean System is a commercial product of Hilt–High Level Software Tools B.V. Clean can be downloaded from its homepage, `http://clean.cs.ru.nl/index.html`.

We do not intend to give a full language reference here, as many information is available in [5] and [15]. We give just a short introduction on how this language (and other functional languages) can be used as a reference language in teaching the implementation of simple algorithms.

Short introduction to Clean language. First we show how simple the usual Hello World! program (see Example 1) is. A one-module Clean program starts with the keyword "module" which is followed by the module name. It must be equivalent with the file name, so if the module name is *helloWorld*, the source code must be saved to *helloWorld.icl*. The second line imports the StdEnv (standard environment), which holds the prototypes of the most important library functions and type definitions.

The Start expression is the replacement of the C-like main function. The evaluation of the Start expression creates the result of the functional program itself. The first program demonstrates how simple the Hello World! program is.

Clean source code

```
module helloWorld
import StdEnv

Start = "Hello World!"
```

Example 1: Hello World!

In the following examples the first two lines are not shown, we should concentrate on the significant lines. In example 2 we assign a numerical expression as the result of the program. In a console application this expression is evaluated and written on the screen.

Clean source code

```
Start = 3-2*4
```

Example 2: Numeric expression

In a functional program we do not have the usual variables, but we can use simple constant value functions similarly to the variables, or constants. In Example 3, *a* and *b* are functions, which evaluate to the values *2* and *3*. As local functions, they are defined inside the Start expression's scope, using the *where* keyword.

Clean source code

```
Start = a+b
  where
    a=2
    b=3
```

Example 3: Using functions

The previous examples have demonstrated how easy the very first steps are in the functional world. Not only simple, but also more complicated types can be used as a result of the program. For example we can define lists easily, and use them as result (see Example 4).

Clean source code

```
Start = resultList
  where
    resultList = [1,3,4,5,6,7,9,4,3,5,6,7,8,4,3]
```

Example 4: Define list

A special datatype called tuple can be found in functional languages. Tuples can be imagined as records without defining its naming the fields. Tuples are values keeping together. We can refer to the values of a tuple by their serial numbers only (1st element, 2nd element, etc.). With tuples we can define functions that return more than one value at a time. In example 5 the Start expression returns a tuple of a string and an integer value. Running this program will show both elements on the screen (separating by commas), writing *3+2=,5*.

Clean source code

```
Start = ("3+2=",3+2)
```

Example 5: Returning tuple

Defining a function usually does not require the description of the function type, it is inferred by the deduction system included in the compiler. We must give a name to the parameter and define the result. In this example we define a function called *increment*, which only has one argument, named *a*. We define the result of this function as *a+1*. The type deduction system will know that the type of the *a* can be anything that the additive operator with an integer value can interpret. We call this function in the Start expression, and give the value of *3* to it. The type deduction system will check if its type (integer) can be added to another integer, and will generate an increment function with this specific type.

Clean source code

```
Start = increment 3

increment a = a+1
```

Example 6: User defined function

Note that calling a function means writing its name and after a space, defining the value for its parameter. There's no need for the C-style function to call operators (parentheses). Nor we use parentheses when a parameter value is a complex expression (see Example 7).

Clean source code

```
Start = decrement (increment 3)

increment a = a+1
decrement a = a-1
```

Example 7: Calling a function

In example 7 we want to evaluate the inner expression first (*increment 3*), then pass its value to function *decrement*.

We can use *patterns* to define different function bodies for different input arguments. In this example we decrement every positive value by one, but decrementing zero means returning the zero value itself (see Example 8).

When we write *decrement 0* (it means " if the first parameter's value equals to 0"), we define the function result as 0. In other cases we define the function result

Clean source code

```
decrement 0 = 0
decrement a = a-1
```

Example 8: Using patterns

as *a-1*. Note that adding a negative value to the function will trigger the second variant of the function body, as a negative value is not equal to 0. We can define different cases for negative values, but not with the pattern match (as we cannot write all the patterns for all the negative numbers), but we can write a *guardian* term (see Example 9).

A guard is a boolean expression that can be inserted between the patterns of a function alternative and the symbol =. The symbol | separates the patterns and the guard. The alternative is only applied when the guards yield True. Each function clause can have a sequence of guarded right-hand sides.

This time we first check if the value of n is less than m or not. If less, we define the function result as the value of m, because it is the maximum of the two numbers. In every other case we define the function result as n, as it holds the maximum value.

Clean source code

```
maximum n m
  | n<m        = m
  | otherwise  = n
```

Example 9: Using guard

In example 9 we used the word *otherwise* and it seems as it would be a keyword for these cases, but it is not. The word *otherwise* is a constant function, always returning with the value of *true*. That means we could write the word *True* instead (see Example 10). Note that we can define more than two cases at the same time, writing more guard terms.

Clean source code

```
maximum n m
  | n<m   = m
  | True  = n
```

Example 10: Using True instead of otherwise

Pattern matching can be applied not only to simple values, but also to lists. In example 11 the first pattern matches to the empty list, and returns zero. The second pattern matches to a list containing one value only. In this case, the parameter named x will holds the value of this element. The third pattern will match to a list which has at least one element. The first element's value will be represented by x in this case, the remaining list goes to *tail*, which means that in this case x is a simple element, and *tail* is a list of elements. Note that the last pattern matches to a one-element list too, and in that case *tail* will be an empty list.

Clean source code

```
count [] = 0
count [x] = 1
count [x:tail] = 2
```

Example 11: Patterns of list

We use the colon operator to add (insert) an element to a list. Note that it will not modify the original list, but will create a brand new list, as function side-effects are denied in functional languages. Function *insert* in Example 12 will insert the parameter value of *a* after the first element of the list by constructing a brand new list.

Clean source code

```
insert a [x:tail] = [x:a:tail]
```

Example 12: Inserting a value into a list

2. Examples of using functional programming methods

Element of a set. Now we will discuss the basic algorithm of *determining if a given value is an element of a set or not.* The set is given as a list of integer values. Normally it is given in imperative algorithm in 13.

Imperative algorithm

```
algorithm isElement
  parameters x:integer, h:list of integer
start
  i:=1
  while i<=length of h and h[i]<>x
     i  := i+1
  end of while
  return (i<=length of h)
end of algorithm
```

Example 13: isElement as imperative algorithm

If the functional paradigm is used to write the previous program, the function can be evaluated in two ways. The first clause is evaluated if the set is empty, so value *x* cannot be element of this empty set. In the second case we separate the set into two parts, an element of the set, and the remains of the set (tail). We can say that x is an element of this set, if it equals to the separated element of the set, or if it is an element of the remaining set (see Example 14). We can use this *isElementOf* function as in the example 15.

Clean source code

```
isElementOf x [] = False
isElementOf x [a:tl]
 | x==a         = True
 | otherwise  = isElementOf x tl
```

Example 14: isElementOf function

Clean source code

```
Start = isElementOf  3 mySet
  where
    mySet  = [1,2,3,5,6,8]
```

Example 15: Use of isElementOf

Counting of elements. Counting the elements can be carried out similarly to the exampe above. The empty list has zero elements, in other cases we can define the length of the list by counting one element at a time (see Example 16).

Clean source code

```
countingElements []   = 0
countingElements [x:tl] = 1 + countingElements tl
```

Example 16: countingElements function

When we want to count those elements only that have a *P property* (in this example, the elements that are *even*), we should modify this function a little by introducing a guard expression. With that we can separate two cases: whether the first element of the list has P property or not (see Example 17).

Clean source code

```
countingElementsEvens []  = 0
countingElementsEvens [x:tl]
  | isEven x   = 1 + countingElementsEvens tl
  | otherwise = countingElementsEvens tl
```

Example 17: counting of P property

Note that the *isEven* function is a library function, and its parameter type must be Int, and its result has to be a boolean value (see Example 18). In Clean, however, we do not need to define the type of a function, in simple cases the type inference system will deduce that. If we want to define a function type explicitly, we can use the double colon after which we can list the types of the input parameters. After the arrow we can give the result type.

Clean source code

```
isEven::Int->Bool
```

Example 18: Prototype of isEven

We can define our own function that has one int parameter only and results a bool, too. In example 19 we define an *isGood* function, which checks if the parameter is in range of $[4 \dots 8)$.

Clean source code

```
isGoodElement::Int->Bool
isGoodElement x = (4<x) && (x<8)
```

Example 19: User defined isGoodElement function

Fortunately, functions in Clean can be passed easily as a parameter, if their names are given. We can define the counting of elements algorithm by using the *P property function as a parameter.* If we want to define the type of *countingElementsAny* function, the first parameter is a list of integers, the second is a function which needs one integer, and returns bool.

Clean source code

```
countingElementsAny:: [Int] (Int->Bool) -> [Int]
countingElementsAny []  isP = 0
countingElementsAny [x:tl] isP
  | isP x         = 1 + countingElementsAny tl isP
  | otherwise = countingElementsAny tl isP
```

Example 20: Defining countingElementsAny function

Giving a function as a parameter is very simple, but its type matches only with one parameter. Without explicitly defining the type of *countingElementsAny*, the type inference system will deduce the same. In the Start expression we can call this function by giving a list and a function as a parameter (see Example 21).

Clean source code

```
Start = countingElementsAny myList isGoodElement
  where
    myList = [1,3,4,5,6,7,9,4,3,5,6,7,8,4,3]
```

Example 21: Calling the countingElementsAny general function

Index of an element. Suppose to have a specific value and a list of values. We need to know what the index of the specific value inside the list is. If it is not in the list, the function must return with 0.

In the first case, the element cannot be found in the empty list, so it returns with 0. In the second case, if the first element equals to the given one, we have its index, and it can return it. Otherwise, we try to determine the index of the value in the remaining list (tail). If we found a good index value (other than zero), we must increase that with 1, because we removed the first element of the tail, and therefore the indices in the tail are shifted by one. If we cannot found the element in the tail, we return with 0 as well (see Example 22).

Clean source code

```
indexOf e [] = 0
indexOf e [x:tl]
  | e==x          = 1
  | index>0       = 1+index
  | otherwise     = 0
  where
     index = indexOf e tl
```

Example 22: Calling the countingElementsAny general function

The maximum of elements. Let's suppose we have a list of integers (a set of integers), and need to determine their maximum value. We give a possible solution for this problem as the *myMaxList* function in example 23. We have chosen this name because a *maxList* function exists in the StdEnv standard library.

Clean source code

```
myMaxList [e]   = e
myMaxList [e:tl]
  | e>max     = e
  | otherwise = max
  where
     max = myMaxList tl
```

Example 23: myMaxList

Sum of elements. Let's suppose we have a list of elements, and we have to determine the sum of these elements. We can define a *genSum* function, which takes a list of integers, and generates the sum of the elements recursively as we show in example 24.

Clean source code

```
genSum [] = 0
genSum [x:tl] = x + genSum tl
```

Example 24: Clean program

Selecting of elements. Let's suppose we have a list of elements, and we need the sublist of the values, gathering the ones with a P property. Let's say we have an *isP* function which can decide whether an element has a P property or not. The solution is very similar to the *countingOfElementsAny* (see in example 20). An empty list has no elements with P properties. Otherwise, if the first element has P property, we will insert it into result before the remaining selected elements, or else it returns the selected elements of the tail (see Example 25).

Clean source code

```
selectingElementsAny []  isP = []
selectingElementsAny [x:tl] isP
  | isP x         = [x : selectingElementsAny tl isP]
  | otherwise = selectingElementsAny tl isP
```

Example 25: Selecting elements

Merging two lists into one. Let's suppose we have two ordered lists, and we have to merge them into one list, keeping the ordering as well. The solution in example 26 handles two different cases. When one of the lists is empty, the result is the another (possibly not the empty) list. Otherwise, when either lists are not empty, we can take the first element of both lists, and decide which is less. If the first element of the first list is the least (named x in the function), insert it into the beginning of the result, and process the remaining lists. The same is the case when the first element of the second list is less.

Clean source code

```
merging [] b  = b
merging a []  = a
merging [x:xtl] [y:ytl]
  | x<y        = [x : merging xtl [y:ytl]]
  | otherwise  = [y : merging [x:xtl] ytl]
```

Example 26: Merging elements

Intersect of two sets. Let's suppose, we have two sets, and we have to determine the intersection of the two sets. We can use the *isElement* function defined above, and a solution is given in example 27.

Clean source code

```
intersect [] _ = []
intersect [x:tl] b
 | isElement x b    = [x : intersect tl b]
 | otherwise    = intersect tl b
```

Example 27: intersection of two sets

The underscore sign in the pattern matches to any value (like the joker char matches to any file name). In this case, we can interpret the first pattern as follows: if the first argument is an empty list, the second argument can be anything. This function can be called from anywhere, as described in example 28.

Quick Sort. We can define the Quick Sort algorithm as in example 29. We use a special list construction mode, which is very close to the mathematical way of giving a set. The `[ x \\ x <- r | x<e ]` means: construct a list of elements x, where x comes from a list named r, and x is less than the value of e. The ++ operator concatenates two lists together.

Clean source code

```
Start = intersect set1 set2
  where
    set1 = [1,2,3,5,6,8]
    set2 = [1,3,4,5,7,8,9]
```

Example 28: Calling intersect

Clean source code

```
qsort [] = []
qsort [e:r] = qsort [ x \\ x <- r | x<e ]
              ++ [e] ++
              qsort [ x \\ x <- r | x>=e ]
```

Example 29: qsort function

3. Conclusion

The most important element of the functional languages is the function. As all functional programs are built up of the composition of functions, it is simple to write example algorithms with the help of functions. Experience shows that students have a strong indisposition for using functions, list expressions or pattern matching. The reason for this is that their way of thinking is based on imperative grounds, and that technology is averse in imperative programs. As the use of guards have grounds in imperative paradigms, they can be easily substituted by a kind of *switch* control structure.

When writing simple functions, the variables, or rather the iterations are missing from the imperative way of thinking. Despite considering them nice and elegant, students do not like using recursive solutions because of their difficulty level. This is so, because they use the already acquired imperative solutions as a starting point, and cannot replace iterations with recursive functions. Most undergraduates find the use of function parameters very exciting, and discover their advantages soon.

After acquiring the functional programming technology, they can view and use algorithms on a higher level of abstraction. This has an impact on their imperative programming style and development. The use of recursive functions often causes problems for beginner programmers, because their training in that field is insufficient. They rarely come across functional thinking in other fields or subjects, like mathematics.

They can hardly get used to regarding a function as a complete prototype while writing it. At the same time, these properties of the functions drive them to learn, as they know and feel that the solution is simple, and they know the principle (the imperative algorithm). They work assiduously on their ideas because they know that they will succeed.

References

[1] Joosten, S., Berg, K., Hoeven, G., *Teaching functional programming to first-year students* Journal of Functional Programming (1993), 3:49–65 Cambridge University Press Cambridge University Press 1993 doi:10.1017/S0956796800000599.

[2] Thompson, S., Wadler, P., *Functional programming in education? Introduction* Journal of Functional Programming (1993), 3:3–4 Cambridge University Press Cambridge University Press 1993 doi:10.1017/S0956796800000563.

[3] Járdán T., Pomaházi S., *Adatszerkezetek és algoritmusok* Liceum Kiadó.

[4] Csőke L., Garamhegyi G., *Adatszerkezetek és algoritmusok* Liceum kiadó.

[5] Nyékiné Gaizler J. (szerk), *Programozási Nyelvek*, Kiskapu Kft, 2003.

[6] Barendregt, H.P., *The lambda Calculus, its Syntax and Semantics*, Amsterdam, North-Holland, 1984.

[7] Csörnyei Zs., *Lambda kalkulus - előadás jegyzet (kézirat).* `http://people.inf.elte.hu/csz/lk-jegyzet.html`.

[8] Horváth, Z., Kozsik, T., *Teaching of Parallel and Distributed Software Design*, Hudák Stefan, Kollár Ján(ed.) in.: Proceedings of the Sixth International Scientific Conference on Electronic Computers and Informatics, ECI 2004 September 22-24 Kosice-Herlány, Slovakia , ISBN.

[9] Porkoláb, Z., Zsók, V., *Teaching Multiparadigm Programming Based on Object-Oriented Programming*, in.: 10th Workshop on Pedagogies and Tools for the Teaching and Learning of Object-Oriented Concepts, TLOOC Workshop, ECOOP 2006, Nantes Nantes, France.

[10] Horváth Z., Kozsik T., Lövei L., *Szoftverrendszerek fejlesztésének oktatása projektfeladat keretén belül* ELTE 2009.

[11] Erlang Consulting. *2008 Obfuscated Erlang Programming Competition homepage.* `http://www.erlang-consulting.com/obfuscatederlang.html`.

[12] Barklund, J., Virding, R., Erlang Reference Manual, 1999. Available from `http://www.erlang.org/download/erl_spec47.ps.gz`. 2007.06.01.

[13] Plasmeijer, R. Eekelen, M., Functional Programming and Parallel Graph Rewriting, Addison-Wesley, 1993.

[14] Achten, P., Wierich, M., A Tutorial to the Clean Object I/O Library, University of Nijmegen, 2000. `http://www.cs.kun.nl/~clean`.

[15] Plasmeijer, R., Eekelen, M., *Clean Language Report v2.1*, `http://clean.cs.ru.nl/download/Clean20/doc/CleanLangRep.2.1.pdf`.

[16] `http://www.haskell.org/`.

[17] `http://msdn.microsoft.com/en-us/library/aa645596`.

Zoltán Hernyák
Roland Király
Eszterházy Károly College
Department of Information Technology
H-3300 Eger, Eszterházy tér 1.
e-mail: `{hz,serial}@aries.ektf.hu`

Annales Mathematicae et Informaticae
36 *(2009) pp. 175–180*
http://ami.ektf.hu

A purely geometric proof of the uniqueness of a triangle with given lengths of one side and two angle bisectors

Victor Oxman

Western Galilee College, Acre, Israel

Submitted 24 September 2009; Accepted 10 November 2009

Abstract

We give a proof of triangle congruence on one side and two angle bisectors based on purely Euclidean geometry methods.

Keywords: Triangle, angle bisector, Steiner-Lehmus theorem

MSC: 51M04, 51M05, 51M25

1. Introduction

In [1, 2] the uniqueness of a triangle with given lengths of one side and two angle bisectors was proven with the help of calculus methods. In this note we give a purely geometric proof of this fact.

2. The uniqueness of a triangle with given lengths of one side and two adjacent angle bisectors

Lemma 2.1. *Suppose that triangles ABC and A'B'C' have an equal side AB=A'B' and equal angle bisectors AL=A'L'. Let* $\angle CAB < \angle C'A'B'$*. Then AC<A'C'.*

Proof. Let $LB = KB, L'B' = K'B'$ (Figure 1). Then $\angle AKB = \angle ALC$ and $\Delta ACL \sim \Delta ABK$, $AC/AB = AL/AK$. Similarly $A'C'/A'B' = A'L'/A'K' = AL/A'K'$. Let $BN \perp AK, B'N' \perp A'K'$. $\angle CAB < \angle C'A'B'$, then $\angle LAB < \angle L'A'B'$ and so $AN > A'N'.AK = 2AN - AL > A'K' = 2A'N' - AL$. Then $AC/AB < A'C'/A'B'$ and $AC < A'C'$. □

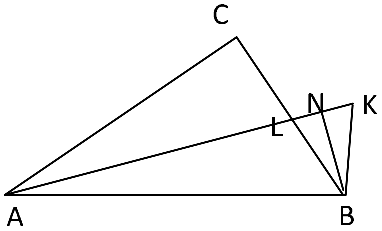


Figure 1

Theorem 2.2. *If one side and two adjacent angle bisectors of a triangle ABC are respectively equal to one side and two adjacent angle bisectors of a triangle A'B'C', then the triangles are congruent.*

Proof. Denote the two angle bisectors of ΔABC by AD and BE and let $AD = A'D', BE = B'E', AB = A'B'$. If $\angle ABC = \angle A'B'C'$, then $\angle ABE = \angle A'B'E' \Rightarrow \Delta ABE \cong \Delta A'B'E' \Rightarrow \angle BAC = \angle B'A'C' \Rightarrow \Delta ABC \cong \Delta A'B'C'$.

Suppose that the triangles ABC and $A'B'C'$ have a common side AB and the adjacent angle bisectors of ΔABC are respectively equal to the adjacent angle bisectors of $\Delta A'B'C'$ $(AD = A'D', BE = B'E')$. We have to consider two cases.

Case 1. $\angle ABC > \angle A'B'C'$ and $\angle BAC > \angle B'A'C'$. Let us suppose that C' is in the interior of the triangle ACF (CF is the altitude of the triangle ACB) or C' is on CF, C' does not coincide with C (see Figure 2). We denote $K = AD \cap CF$ and

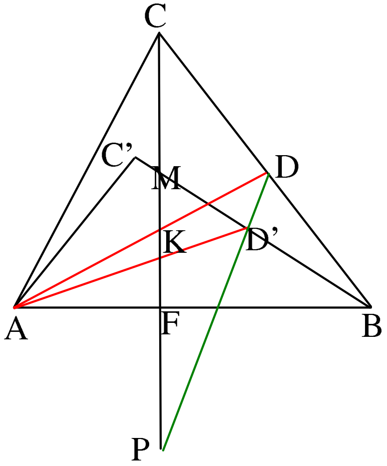


Figure 2

$M = C'B \cap CF$.

$$AC' < AC \Rightarrow \frac{AC}{AB} = \frac{CD}{DB} > \frac{AC'}{AB} = \frac{C'D'}{D'B} \geqslant \frac{MD'}{D'B} \quad ,$$

so $(DD') \cap (CF) = P$ and M is an interior point of interval CP. $\Delta DAD'$ is isosceles and therefore $\angle KD'P > 90°$, but $90° > \angle AKF > \angle KD'P$ and so we have a contradiction with $\angle KD'P > 90°$. So C' can not be in the interior of the triangle ACF or on CF. Similarly we get that C' can not be in the interior of the triangle BCF.
So the Case 1 is impossible.
Case 2. $\angle ABC < \angle A'B'C'$ and $\angle BAC > \angle B'A'C'$ (Figure 3).

We have $AC > AC'$ and $BC' > BC$ (Lemma 2.1). So $\angle CC'A > \angle ACC'$ and $\angle C'CB > \angle CC'B$. But $\angle ACC' > \angle C'CB$ and $\angle CC'B > \angle CC'A$. Then we again get a contradiction and this case is impossible too. □

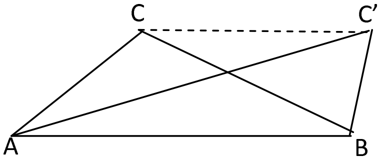


Figure 3

3. The uniqueness of a triangle with given lengths of one side, one adjacent angle bisector and the opposite angle bisector

Lemma 3.1. *Suppose that triangles ABC and A'B'C' have an equal side AB=A'B' and equal angle bisector AL=A'L'. Let* $\angle BAC < \angle B'A'C'$*. Then BC<B'C'.*

Proof. By Lemma 2.1 we get $AC < A'C'$. Let $BH \perp AC$ and $B'H' \perp A'C'$ (Figure 4). So $AH > A'H'$and $BH < B'H'$. Then $CH = |AH - AC| < C'H' =$

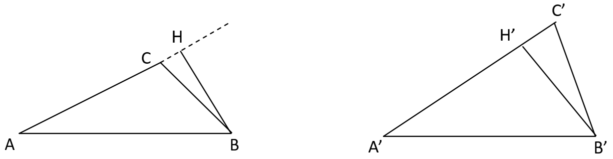


Figure 4

$|A'H' - A'C'|$ and so we have two right-angled triangles CHB and $C'H'B'$ with $CH < C'H'$ and $BH < B'H'$. Let $H'F = HC$ and $H'K = HB$ (Figure 5). So

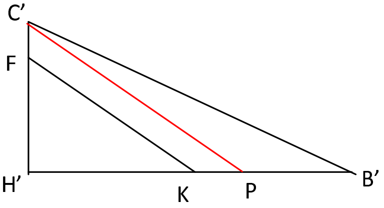


Figure 5

$FK = CB$. If $FK||C'B'$, then $FK < C'B'$. Suppose $\angle FKH' > \angle C'B'H'$. Let $C'P||FK$. Then $C'P > FK$. $\angle C'PB'$ is an obtuse angle and so $C'B' > C'P > FK = CB$. □

Theorem 3.2. *If one side, one adjacent angle bisector and the opposite angle bisector of a triangle ABC are respectively equal to one side, one adjacent angle bisector and the opposite angle bisector of a triangle A'B'C', then the triangles are congruent.*

Proof. Denote the two angle bisectors of triangles ABC and $A'B'C'$ by $AD, A'D'$ and $CE, C'E'$ correspondently and let $AD = A'D', CE = C'E', AB = A'B'$. Similarly to the proof of Theorem 2.2 we conclude that if $\angle BAC = \angle B'A'C'$ then the triangles are congruent. Let $\angle BAC < \angle B'A'C'$, then $A'C' > AC$ and $C'B' > CB$ (Lemma 2.1, 3.1). We prove that $C'E' > CE$. Let $\angle B"A'D' = \angle C"A'D' = \angle BAD, A'B" = AB, A'C" = AC$ (Figure 6), then $\Delta B"A'C' \cong \Delta BAC$ ($A'D'$ is a common angle bisector of the triangles $B'A'C'$ and $B"A'C"$).

We have to consider 3 cases.

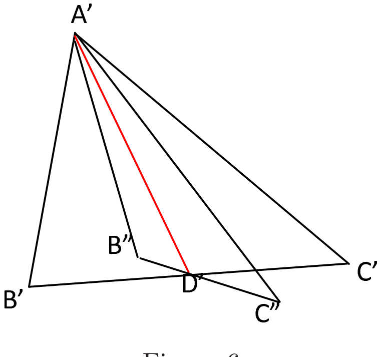


Figure 6

Case 1. Point $C"$ is in the interior of $\Delta C'A'D'$ (include interval $D'C'$).

In [3 ,Theorem 3] it was proven that in this case $C"E" = CE < C'E'$.
Case 2. Point $C"$ is in the exterior of $\Delta C'A'D'$ and $\angle A'C"B" = \angle ACB > \angle A'C'B'$. Let $C_1A' = CA$, $\angle A'C_1B_1 = \angle ACB$, $\angle C_1A'B_1 = \angle CA'B$
(Figure 7).

So $\Delta C_1A'B_1 \cong \Delta CAB$. According to [3, Lemma 1] the bisector of $\angle A'C_1B_1$

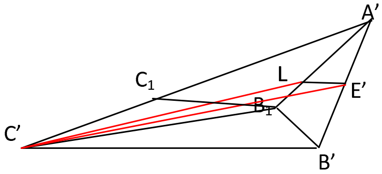


Figure 7

is less than the bisector of $\angle A'C'B_1$. Let $C'L$ be the triangle $A'C'B_1$ bisector. $\angle B_1A'C' < \angle B'A'C'$, so $C'B_1 < C'B'$ (the purely geometric proof of this fact was given in Euclid's Elements, Book 1, proposition 24). Then $B_1L/LA' = C'B_1/C'A' < C'B'/C'A' = B'E'/E'A'$. $A'B_1 = A'B'$ and so $\angle B_1LE'$ is an obtuse angle and $\angle C'LE' > \angle B_1LE' > 90°$. Then $C'E' > C'L > CE$.
Case 3. Point $C"$ is in the exterior of $\Delta C'A'D'$ and $\angle A'C"B" = \angle ACB < \angle A'C'B'$. Let $C'B_2||C_1B_1$ and let $C'L_1$ be the angle bisector of the triangle $A'C'B_2$ (Figure 8). Then $C'L_1 > CE$. $C'B_2 < C'B'$ and again $B_2L_1/L_1A' = C'B_2/C'A' < C'B'/C'A' = B'E'/E'A'$, $\angle C'L_1E'$ is an obtuse angle and $C'E' > C'L_1 > CE$. □

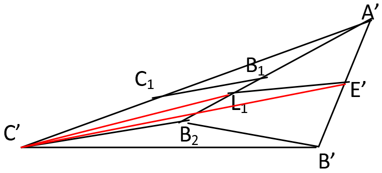


Figure 8

4. Notes

From each one of Theorem 2.1 and of [3,Theorem 3] the Steiner-Lehmus Theorem obviously follows and so these theorems provide its pure geometric proof.

References

[1] Oxman,V., On the existence of triangles with given lengths of one side and two adjacent angle bisectors, *Forum Geom.*, 4 (2004) 215–218.

[2] Oxman,V., On the Existence of Triangles with Given Lengths of One Side, the Opposite and One Adjacent Angle Bisectors, *Forum Geom.*, 5 (2005) 21–22.

[3] Oxman,V., A Purely Geometric Proof of the Uniqueness of a Triangle With Prescribed Angle Bisectors, *Forum Geom.*, 8 (2008) 197–200.

Victor Oxman
Western Galilee College, Acre, Israel
e-mail: `victor.oxman@gmail.com`