Contents

Methodological papers

ANNALES MATHEMATICAE ET INFORMATICAE 35. (2008)

ANNALES MATHEMATICAE ET INFORMATICAE

TOMUS 35. (2008)

HUNGARIA, EGER

ANNALES MATHEMATICAE ET INFORMATICAE

VOLUME 35. (2008)

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: 2008. december Példányszám: 50

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

Annales Mathematicae et Informaticae
35 *(2008) pp. 3–10*
http://www.ektf.hu/ami

Laplace transform pairs of N-dimensions and second order linear partial differential equations with constant coefficients

A. Aghili, B. Salkhordeh Moghaddam

Department of Mathematics, Faculty of Science
University of Guilan, Rasht, Iran

Submitted 5 February 2008; Accepted 15 September 2008

Abstract

In this paper, authors will present a new theorem and corollary on multi-dimensional Laplace transformations. They also develop some applications based on this results. The two-dimensional Laplace transformation is useful in the solution of partial differential equations. Some illustrative examples related to Laguerre polynomials are also provided.

Keywords: Two-dimensional Laplace transforms, second-order linear non-homogenous partial differential equations, Laguerre polynomials.

MSC: 44A30, 35L05

1. Introduction

In [3] R. S. Dahiya established several new theorems for calculating Laplace transform pairs of N-dimensions and two homogenous boundary value problems related to heat equations were solved. In [4] J. Saberi Najafi and R. S. Dahiya established several new theorems for calculating Laplace transforms of n-dimensions and in the second part application of those theorems to a number of commonly used special functions was considered, and finally, by using two dimensional Laplace transform, one-dimensional wave equation involving special functions was solved. Later in [1, 2] authors, established new theorems and corollaries involving systems of two-dimensional Laplace transforms containing several equations.

The generalization of the well-known Laplace transform

$$L[f(t);s] = \int_0^\infty e^{-st} f(t)\,\mathrm{d}t,$$

to n-dimensional is given by

$$L_n[f(\bar{t});\bar{s}] = \int_0^\infty \int_0^\infty \cdots \int_0^\infty \exp(-\bar{s}\bar{t}) f(\bar{t}) P_n(\mathrm{d}\bar{t}),$$

where $\bar{t} = (t_1, t_2, \ldots, t_n)$, $\bar{s} = (s_1, s_2, \ldots, s_n)$, $\bar{s}\bar{t} = \sum_{i=1}^n s_i t_i$ and $P_n(\mathrm{d}\bar{t}) = \prod_{k=1}^n \mathrm{d}t_k$. In addition to the notations introduced above, we will use the following throughout this paper.

Let $\bar{t^v} = (t_1^v, t_2^v, \ldots, t_n^v)$ for any real exponent v and let $P_k(\bar{t})$ be the k-th symmetric polynomial in the components t_k of $\bar{t}$. Then

$$\begin{aligned}
P_0(\bar{t^v}) &= 1,\\
P_1(\bar{t^v}) &= t_1^v + t_2^v + \ldots + t_n^v,\\
P_2(\bar{t^v}) &= \sum\nolimits_{i,j=1,i<j}^n t_i^v t_j^v,\\
&\vdots\\
P_n(\bar{t^v}) &= t_1^v t_2^v \ldots t_n^v.
\end{aligned}$$

The inverse Laplace transform is given by

$$L^{-1}[F(\bar{s});\bar{t}] = \left(\frac{1}{2i\pi}\right)^n \int_{a-i\infty}^{a+i\infty} \int_{d-i\infty}^{d+i\infty} \cdots \int_{c-i\infty}^{c+i\infty} e^{-\bar{s}\bar{t}} F(\bar{s}) P_n(\bar{s})\, \mathrm{d}\bar{s}.$$

2. The main theorem

Theorem 2.1. *Let*

$$g(s) = L[f(t);s], \quad F(s) = L[t^{-3/2} g(1/t);s], \quad H(s) = L[tf(t^4);s].$$

If $f(t)$, $t^{-3/2} g(\frac{1}{t})$ and $tf(t^4)$ are continuous and integrable on $(0,\infty)$, then

$$L_n\left[P_n(\bar{t}^{-1/2}) F\left(P_1^2(\bar{t}^{-1})\right);\bar{s}\right] = 4\pi^{(n+1)/2} \frac{H[2\sqrt{2}P_1(\bar{s}^{1/2})]}{P_n(\bar{s}^{1/2})},$$

where $n = 1, 2, \ldots, N$.

Proof. We have

$$g\left(\frac{1}{t}\right) = \int_0^\infty \exp\left(-\frac{u}{t}\right) f(u)\, \mathrm{d}u. \tag{2.1}$$

Multiply both sides of (2.1) by $t^{-3/2}\exp(-st)$, $\mathrm{Re}(s) > 0$ and integrate with respect to t on $(0,\infty)$ to get

$$\int_0^\infty \frac{e^{-st} g\left(t^{-1}\right)}{t^{3/2}}\, \mathrm{d}t = \int_0^\infty \int_0^\infty e^{-st} e^{-\frac{u}{t}} f\left(u\right) t^{-3/2}\, \mathrm{d}u\, \mathrm{d}t. \tag{2.2}$$

Since the integral on the right side of (2.2) is absolutely convergent, we may change the order of integration to obtain

$$\int_0^\infty \frac{e^{-st} g\left(t^{-1}\right)}{t^{3/2}}\,\mathrm{d}t = \int_0^\infty f(u) \int_0^\infty e^{-st-u/t} t^{-3/2}\,\mathrm{d}t\,\mathrm{d}u. \tag{2.3}$$

Evaluating the inner integral on the right side of (2.3), we get

$$F(s) = \sqrt{\pi} \int_0^\infty \frac{f(u)\, e^{-\sqrt{su}}}{\sqrt{u}}\,\mathrm{d}u.$$

Now, on setting $u = v^4$, replacing s by $P_1^2(\bar{t}^{-1})$ and then multiplying both sides of (2.3) by $P_n(\bar{t}^{-1/2})e^{-s\bar{t}}$ and integrating with respect to $t_1, t_2, \ldots, t_n$ from 0 to ∞, leads to the statement. □

Corollary 2.2. *Letting $n = 2$ we get from Theorem 2.1, that*

$$L_2\left\{\frac{1}{\sqrt{xy}} F\left(\left(\frac{1}{x}+\frac{1}{y}\right)^2\right); u, v\right\} = 4\pi^{3/2} \frac{H[2\sqrt{2}(\sqrt{u}+\sqrt{v})]}{\sqrt{uv}}. \tag{2.4}$$

As an application of the above theorem and corollary, some illustrative examples in two dimensions are also provided.

Example 2.3. Let $f(t) = \sin(\sqrt{t})$, then $F(s) = \frac{2\sqrt{\pi}}{1+4s}$,

$$H(s) = \frac{1}{2} + \frac{2\sqrt{\pi}}{8}\left\{s\cos\left(\frac{s^2}{4}\right)\left(2S\left(\frac{s}{2\sqrt{\pi}}\right)-1\right) + s\sin\left(\frac{s^2}{4}\right)\left(1-2C\left(\frac{s}{2\sqrt{\pi}}\right)\right)\right\},$$

hence

$$\begin{aligned} &L_2\left[\frac{(xy)^{\frac{3}{2}}}{4(x+y)^2+x^2y^2}, u, v\right] \\ &= \sqrt{\frac{\pi}{uv}}\left\{\pi(\sqrt{u}+\sqrt{v})\cos\left(2(\sqrt{u}+\sqrt{v})^2\right)\left(2S\left(\frac{2(\sqrt{u}+\sqrt{v})}{\sqrt{\pi}}\right)-1\right)\right. \\ &\left.+(\sqrt{u}+\sqrt{v})\sin\left(2(\sqrt{u}+\sqrt{v})^2\right)\left(1-2C\left(\frac{2(\sqrt{u}+\sqrt{v})}{\sqrt{\pi}}\right)\right)+\sqrt{\pi}\right\}, \end{aligned}$$

where Fresnel's integrals are defined as following

$$C(x) = \frac{1}{\sqrt{2\pi}}\int_0^x \frac{\cos(t)}{\sqrt{t}}\,\mathrm{d}t, \quad S(x) = \frac{1}{\sqrt{2\pi}}\int_0^x \frac{\sin(t)}{\sqrt{t}}\,\mathrm{d}t.$$

Example 2.4. If $f(t) = \ln(\alpha t)$ then

$$F(s) = \frac{1}{s}\{\ln(\alpha/s) - \gamma\} \quad \text{and} \quad H(s) = \frac{1}{s}\{\ln(\alpha) + 4(1-\gamma-\ln(s))\}.$$

Using (2.4), we arrive at

$$L_2\left[\frac{\sqrt{xy}}{x+y}\left(\ln\left(\frac{4(x+y)^2}{\alpha(xy)^2}\right)-2\gamma\right),u,v\right]$$
$$=\pi\frac{4\ln(\sqrt{u}+\sqrt{v})-\ln(\alpha)+6\ln(2)+4(\gamma-1)}{2\sqrt{uv}(\sqrt{u}+\sqrt{v})^2}.$$

In the following example, we give an application of two-dimensional Laplace transforms and complex inversion formula for calculating some of the series related to Laguerre polynomials.

Example 2.5. We shall show that (see [6])

1. $\sum_{n=0}^{\infty}L_n(x)L_n(y)\lambda^n=\frac{1}{1-\lambda}e^{-\frac{\lambda(x+y)}{1-\lambda}}I_0\left(\frac{2\sqrt{\lambda xy}}{1-\lambda}\right)$,
2. $\sum_{n=0}^{\infty}L_n(t)L_n(\xi)=e^t\delta(t-\xi)$,

where $L_n(x)$ is Laguerre polynomial and $I_0(x)$ is modified Bessel 's function of order zero.

Solution.

1. It is well known that $L[L_n(x),p]=\frac{1}{p}\left(1-\frac{1}{p}\right)^n$. Taking two-dimensional Laplace transform of the left hand side, leads to the following

$$L_2\left[\sum_{n=0}^{\infty}L_n(x)L_n(y)\lambda^n,p,q\right]=\int_0^{\infty}\int_0^{\infty}\left(\sum_{n=0}^{\infty}L_n(x)L_n(y)\lambda^n e^{-px-qy}\right)\mathrm{d}x\,\mathrm{d}y.$$

Changing the order of summation and double integration to get

$$L_2\left[\sum_{n=0}^{\infty}L_n(x)L_n(y)\lambda^n,p,q\right]=\sum_{n=0}^{\infty}\int_0^{\infty}\int_0^{\infty}L_n(x)L_n(y)\lambda^n e^{-px-qy}\,\mathrm{d}x\,\mathrm{d}y.$$

The value of the inner integral is

$$\sum_{n=0}^{\infty}\lambda^n\int_0^{\infty}\int_0^{\infty}L_n(x)L_n(y)e^{-px-qy}\,\mathrm{d}x\,\mathrm{d}y$$
$$=\sum_{n=0}^{\infty}\lambda^n\left\{\frac{1}{pq}\left(1-\frac{1}{p}\right)^n\left(1-\frac{1}{q}\right)^n\right\}=\frac{1}{1-\lambda}\frac{1}{pq+k(p+q)-k},$$

where $k=\frac{\lambda}{1-\lambda}$. Using complex inversion formula for two-dimensional Laplace transform to obtain,

$$\sum_{n=0}^{\infty}L_n(x)L_n(y)\lambda^n$$
$$=\left(\frac{1}{2i\pi}\right)^2\int_{a-i\infty}^{a+i\infty}\int_{d-i\infty}^{d+i\infty}e^{px+qy}\frac{1}{1-\lambda}\frac{1}{pq+k(p+q)-k}\,\mathrm{d}p\,\mathrm{d}q$$

$$= \frac{1}{1-\lambda}\frac{1}{2i\pi}\int_{a-i\infty}^{a+i\infty}\left\{\frac{1}{2i\pi}\int_{d-i\infty}^{d+i\infty}\frac{e^{px}}{pq+k(p+q)-k}dp\right\}e^{qy}\,\mathrm{d}q$$

$$= \frac{1}{1-\lambda}\frac{1}{2i\pi}\int_{a-i\infty}^{a+i\infty}\frac{e^{-\frac{kx(q-1)}{q+k}}}{q+k}e^{qy}\,\mathrm{d}q = \frac{1}{1-\lambda}e^{-\frac{\lambda(x+y)}{1-\lambda}}I_0\left(\frac{2\sqrt{\lambda xy}}{1-\lambda}\right).$$

2. Taking two-dimensional Laplace transform of the left hand side, leads to the following

$$L_2\left[\sum_{n=0}^{\infty}L_n(t)L_n(\xi),p,q\right] = \int_0^{\infty}\int_0^{\infty}\left(\sum_{n=0}^{\infty}L_n(t)L_n(\xi)e^{-pt-q\xi}\right)\mathrm{d}t\,\mathrm{d}\xi.$$

Changing the order of summation and double integration to get,

$$L_2\left[\sum_{n=0}^{\infty}L_n(t)L_n(\xi),p,q\right] = \sum_{n=0}^{\infty}\int_0^{\infty}\int_0^{\infty}L_n(t)L_n(\xi)e^{-pt-q\xi}\,\mathrm{d}t\,\mathrm{d}\xi.$$

It is not difficult to show that the value of the inner integral is

$$\int_0^{\infty}\int_0^{\infty}L_n(t)L_n(\xi)e^{-pt-q\xi}\,\mathrm{d}t\,\mathrm{d}\xi = \frac{1}{pq}\left(1-\frac{1}{p}\right)^n\left(1-\frac{1}{q}\right)^n$$

and

$$\sum_{n=0}^{\infty}\frac{1}{pq}\left(1-\frac{1}{p}\right)^n\left(1-\frac{1}{q}\right)^n = \frac{1}{p+q-1}.$$

Using complex inversion formula for two-dimensional Laplace transforms to obtain,

$$\sum_{n=0}^{\infty}L_n(t)L_n(\xi) = \left(\frac{1}{2i\pi}\right)^2\int_{a-i\infty}^{a+i\infty}\int_{b-i\infty}^{b+i\infty}\frac{e^{pt+q\xi}}{p+q-1}\,\mathrm{d}p\,\mathrm{d}q.$$

The above double integral may be re-written as follows,

$$\sum_{n=0}^{\infty}L_n(t)L_n(\xi) = \frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty}e^{q\xi}\left\{\frac{1}{2\pi i}\int_{b-i\infty}^{b+i\infty}\frac{e^{pt}}{p-(1-q)}\,\mathrm{d}p\right\}\mathrm{d}q.$$

The value of the inner integral by residue theorem is equal to $e^{(1-q)t}$, upon substitution of this value in double integral we get,

$$\sum_{n=0}^{\infty}L_n(t)L_n(\xi) = \frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty}e^{q\xi}e^{(1-q)t}\,\mathrm{d}q = e^t\frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty}e^{-q(t-\xi)}\,\mathrm{d}q,$$

therefore

$$\sum_{n=0}^{\infty}L_n(t)L_n(\xi) = e^t\delta(t-\xi).$$

3. Solution to second-order linear partial differential equations with constant coefficients

The general form of second-order linear partial differential equation in two independent variables is given by (see [5]).

$$Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = q(x,y), \quad 0 < x, y < \infty, \tag{3.1}$$

where A, B, C, D, E and F are given constant and $q(x,y)$ is source function of x and y or constant. We will use the following for the rest of this section (see [5, 6]). If

$$\begin{aligned} &u(x,0) = f(x), \quad u(0,y) = g(y), \quad u_y(x,0) = f_1(x), \\ &u_x(0,y) = g_1(y), \quad u(0,0) = u_0 \end{aligned} \tag{3.2}$$

and if their one-dimensional Laplace transformations are $F(u)$, $G(v)$, $F_1(u)$ and $G_1(v)$, respectively, then

$$\begin{aligned} &L_2[u(x,y);u,v] = \int_0^\infty \int_0^\infty u(x,t)e^{-ux-vt}\,\mathrm{d}x\,\mathrm{d}t = U(u,v), \\ &L_2[u_{xx};u,v] = u^2U(u,v) - uG(v) - G_1(v), \\ &L_2[u_{xy};u,v] = uvU(u,v) - uF(u) - vG(v) - u(0,0), \\ &L_2[u_{yy};u,v] = v^2U(u,v) - uF(u) - F_1(u), \\ &L_2[u_x;u,v] = uU(u,v) - G(v), \\ &L_2[u_y;u,v] = vU(u,v) - F(u). \end{aligned} \tag{3.3}$$

Applying double Laplace transformation term wise to partial differential equations and the initial-boundary conditions in (3.2) and using (3.3), we obtain the transformed problem

$$\begin{aligned} U(u,v) = &\frac{1}{Au^2 + Cv^2 + Buv + Ev + Du + F}\{A(uG(v) + G_1(v)) \\ &+ B(uF(u) + vG(v) - u_0) + C(vF(u) + F_1(u)) \\ &+ DG(v) + EF(u) + Q(u,v)\}. \end{aligned} \tag{3.4}$$

Now, in the following examples we illustrate the above method.

Example 3.1. Letting $A = B = C = 0$, we get

$$Du_x + Eu_y + Fu = q(x,y), \quad 0 < x, y < \infty, \quad (E/D > 0).$$

With initial boundary conditions

$$u(x,0) = f(x), \quad u(0,y) = g(y),$$

application of the relationship (3.4) gives

$$U(u,v)=\frac{DG(v)+EF(u)+Q(u,v)}{Ev+Du+F}. \tag{3.5}$$

The inverse double Laplace transform of (3.5) leads to the formal solution

$$u(x,y)=e^{-\frac{F}{D}x}g\left(y-\frac{E}{D}x\right)+e^{-\frac{F}{E}y}f\left(x-\frac{D}{E}y\right)$$
$$+\begin{cases}\frac{1}{D}\int_0^x e^{-\frac{F}{D}\xi}q(x-\xi,y-\frac{E}{D}\xi)\,\mathrm{d}\xi, & \text{if } y>\frac{E}{D}x,\\ \frac{1}{E}\int_0^y e^{-\frac{F}{E}\eta}q(x-\frac{D}{E}\eta,y-\eta)\,\mathrm{d}\eta, & \text{if } y<\frac{E}{D}x.\end{cases}$$

Example 3.2. If $C=E=D=0$, $A=1$, $B=\alpha$, $F=\beta$, then (3.1) reduces to

$$u_{xx}+\alpha u_{xy}+\beta u=q(x,y),\quad 0<x,y<\infty.$$

With the following initial conditions

$$u(0,y)=g(y),\quad u_x(0,y)=g_1(y),\quad u(x,0)=0,\quad u(0,0)=u_0$$

we obtain

$$U(u,v)=\frac{1}{u^2+\alpha uv+\beta}\{uG(v)+G_1(v)+\alpha(vG(v)-u_0)+Q(u,v)\}. \tag{3.6}$$

The inverse double Laplace transform of (3.6) yields (see [7])

$$u(x,y)=L_2^{-1}[U(u,v)]=L_2^{-1}\left[\frac{Q(u,v)}{u^2+\alpha uv+\beta}\right]+L_2^{-1}\left[\frac{uG(v)}{u^2+\alpha uv+\beta}\right]$$
$$+L_2^{-1}\left[\frac{G_1(v)}{u^2+\alpha uv+\beta}\right]+\alpha L_2^{-1}\left[\frac{vG(v)}{u^2+\alpha uv+\beta}\right]+\alpha u_0L_2^{-1}\left[\frac{1}{u^2+\alpha uv+\beta}\right]$$

or equivalently

$$u(x,y)=\int_0^x\int_0^\xi J_0\left(2\sqrt{\beta\eta(x-\xi)}\right)q(\xi-\eta,y-\alpha\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi$$
$$+g(y-\alpha x)+\frac{1}{\alpha}\int_0^{\alpha x}\sqrt{\frac{\beta\eta}{\alpha x-\eta}}J_1\left(2\sqrt{\frac{\beta\eta}{\alpha}(x-\frac{\eta}{\alpha})}\right)g(y-\eta)\,\mathrm{d}\eta$$
$$+\frac{1}{\alpha}\int_0^{\alpha x}J_0\left(2\sqrt{\frac{\beta\eta}{\alpha}(x-\frac{\eta}{\alpha})}\right)g_1(y-\eta)\,\mathrm{d}\eta+g(y)-g(y-\alpha x)$$
$$+\frac{1}{\alpha}\int_0^{\alpha x}\sqrt{\frac{\beta\eta}{\alpha x-\eta}}\left(2-\frac{\alpha x}{\eta}\right)J_1\left(2\sqrt{\frac{\beta\eta}{\alpha}(x-\frac{\eta}{\alpha})}\right)g(y-\eta)\,\mathrm{d}\eta$$
$$+\begin{cases}0, & \text{if } y>\alpha x,\\ \alpha u_0J_0(\frac{2}{\alpha}\sqrt{\beta y(\alpha x-\eta)}), & \text{if } y<\alpha x.\end{cases}$$

4. Conclusions

The multi-dimensional Laplace transform provides powerful method for analyzing linear systems. It is heavily used in solving differential and integral equations. The main purpose of this work is to develop a method of computing Laplace transform pairs of N-dimensions from known one-Dimensional Laplace transform and making continuous effort in expanding the transform tables and in designing algorithms for generating new inverses and direct transform from known ones. It is clear that the theorems of that type described here can be further generated for other type of functions and relations. These relations can be used to calculate new Laplace transform pairs.

Acknowledgements. The authors would like to thank referees for their comments and questions.

References

[1] Aghili, A., Salkhordeh Moghaddam, B., Laplace transform pairs of n-dimensions and a Wave equation, *Intern. Math. Journal*, 5(4) (2004) 377–382.

[2] Aghili, A., Salkhordeh Moghaddam, B., Multi-dimensional laplace transform and systems of partial differential equations, *Intern. Math. Journal.*, 1 (2006) 21–24.

[3] Dahiya, R.S., Vinayagamoorty, M., Laplace transform pairs of N-dimensions and heat conduction problem, *Math. Comput. Modelling.*, 10 (13) (1990) 35–50.

[4] Dahiya, R.S., Saberi-Nadjafi, J., Theorems on N-dimensional laplace transforms and their applications, *15th annual Conference of Applied Mathematics, Univ. of Central Oklahoma, Electronic Journal of Differential Equations*, 02 (1999) 61–74.

[5] Ditkin, V.A., Prudnikov, A.P., Operational calculus in two variables and its application, *New York*, (1962).

[6] Roberts, G.E., Kaufman, H., Table of taplace transforms, *Philadelphia, W. B. Saunders Co.*, (1966).

A. Aghili
B. Salkhordeh Moghaddam
Department of Mathematics
Faculty of Sciences
Namjoo St., Rasht
Iran

e-mail:
armanaghili@yahoo.com
salkhorde@yahoo.com

Annales Mathematicae et Informaticae
***35** (2008) pp. 11–19*
http://www.ektf.hu/ami

On positive integers with a certain nondivisibility property*

Ioulia Baoulina[a], Florian Luca[b]

[a]Harish-Chandra Research Institute

[b] Instituto de Matemáticas, Universidad Nacional Autonoma de México

Submitted 25 May 2008; Accepted 5 September 2008

Abstract

For a positive integer $k \geqslant 3$ let $(u_m^{(k)})_{m\geqslant 0}$ be the Lucas sequence given by $u_0^{(k)} = 0,\ u_1^{(k)} = 1$ and $u_{m+2}^{(k)} = ku_{m+1}^{(k)} - u_m^{(k)}$ for all $m \geqslant 0$. In this paper, we study the positive integers n such that

$$\frac{n-k}{1+(k-2)(u_m^{(k)})^2} \notin \mathbb{Z} \qquad \text{for any } 3 \leqslant k < n \text{ and } m \geqslant 1.$$

Keywords: Diophantine Equations, Primes, Euler Function, Fibonacci Numbers

MSC: 11N25, 11N36

1. Introduction

For a positive integer $k \geqslant 3$ let $(u_m^{(k)})_{m\geqslant 0}$ be the Lucas sequence given by $u_0^{(k)} = 0,\ u_1^{(k)} = 1$ and $u_{m+2}^{(k)} = ku_{m+1}^{(k)} - u_m^{(k)}$ for all $m \geqslant 0$. In this paper, we study the positive integers n such that

$$\frac{n-k}{1+(k-2)(u_m^{(k)})^2} \notin \mathbb{Z} \qquad \text{for any } 3 \leqslant k < n \text{ and } m \geqslant 1. \tag{1.1}$$

Let $\mathcal{N}$ be the set of positive integers satisfying property (1.1). The study of this set of integers is motivated by the study of the solutions of the Diophantine equation

$$x_1^2 + \cdots + x_n^2 = yx_1 \cdots x_n, \quad n \geqslant 3, \tag{1.2}$$

*We thank the referee for suggestions that improved the quality of this paper.

in positive integers $x_1, \dots, x_n, y$. Hurwitz [5], proved that the Diophantine equation (1.2) has no solutions with $y > n$ and has infinitely many solutions with $y = n$. Herzberg [4], showed that there are only 15 values of $n \leqslant 301020$ for which (1.2) has no solutions with $y < n$. In particular, for any $2688 < n \leqslant 301020$, equation (1.2) has solutions with $y < n$. Using Herzberg's algorithm, we checked all $n \leqslant 10^8$ and didn't find any other exceptional values. It is conjectured that for a sufficiently large n, equation (1.2) has a solution with $y < n$. Let us remark that Hurwitz's results yield that $(u_{m+1}^{(k)} - u_m^{(k)}, u_m^{(k)} - u_{m-1}^{(k)}, \underbrace{1, \dots, 1}_{k-2}, k)$ is a solution of the equation

$$y_1^2 + \cdots + y_k^2 = z y_1 \cdots y_k$$

for any $k \geqslant 3$ and $m \geqslant 1$. It is easy to check that

$$(u_{m+1}^{(k)} - u_m^{(k)})(u_m^{(k)} - u_{m-1}^{(k)}) = 1 + (k-2)(u_m^{(k)})^2.$$

Hence, if for a given n there exist $3 \leqslant k < n$ and $m \geqslant 1$ such that $\dfrac{n-k}{1+(k-2)(u_m^{(k)})^2}$ is an integer, then $(u_{m+1}^{(k)} - u_m^{(k)}, u_m^{(k)} - u_{m-1}^{(k)}, \underbrace{1, \dots, 1}_{n-2}, y)$ is a solution of (1.2), where

$$\begin{aligned} y &= \frac{(u_{m+1}^{(k)} - u_m^{(k)})^2 + (u_m^{(k)} - u_{m-1}^{(k)})^2 + k - 2}{(u_{m+1}^{(k)} - u_m^{(k)})(u_m^{(k)} - u_{m-1}^{(k)})} + \frac{n-k}{1+(k-2)(u_m^{(k)})^2} \\ &= k + \frac{n-k}{1+(k-2)(u_m^{(k)})^2} < n. \end{aligned}$$

In particular, if for any sufficiently large n we could find such values of k and m, then the conjecture would follow. Unfortunately, there are infinitely many values of n which are in the set $\mathcal{N}$, and this is the content of our paper.

2. Result

Our precise result is the following. For a set $\mathcal{A}$ of positive integers and a positive real number x let $\mathcal{A}(x) = \mathcal{A} \cap [1, x]$.

Theorem 2.1. *There exists x_0 such that $\#\mathcal{N}(x) > 0.09x/\log x$ for $x > x_0$.*

For the proof, we will need the following lemma. For a positive integer m let $\phi(m)$ denote the Euler function of m.

Lemma 2.2. *We have the estimate*

$$S = \sum_{k \geqslant 3} \sum_{m \geqslant 2} \frac{1}{\phi(1+(k-2)(u_m^{(k)})^2)} < 0.91. \tag{2.1}$$

Proof. Let $\omega(m)$ be the number of distinct prime factors of the positive integer m. Thus, if $p_1 < p_2 < \cdots < p_{\omega(m)}$ denote all the prime factors of $m > 1$, then

$$\frac{\phi(m)}{m} = \prod_{i=1}^{\omega(m)} \left(1 - \frac{1}{p_i}\right) \geqslant \prod_{i=1}^{\omega(m)} \left(1 - \frac{1}{i+1}\right) = \frac{1}{\omega(m)+1}.$$

From here, we can deduce various things. For example, since $m \geqslant 2^{\omega(m)}$, we get that $\omega(m) \leqslant (\log m)/(\log 2)$, therefore the above inequality gives

$$\frac{\phi(m)}{m} \geqslant \frac{1}{(\log m)/(\log 2) + 1} = \frac{\log 2}{\log(2m)}. \tag{2.2}$$

Then

$$\frac{1}{\phi(m)} \leqslant \frac{\log(2m)}{m \log 2}.$$

Applying this to $1 + (k-2)(u_m^{(k)})^2$, we get

$$\frac{1}{\phi(1 + (k-2)(u_m^{(k)})^2)} \leqslant \frac{\log(2(1 + (k-2)(u_m^{(k)})^2))}{(\log 2)(1 + (k-2)(u_m^{(k)})^2)}.$$

For $m \geqslant 2$ and $k \geqslant 3$ we have that

$$1 + (k-2)(u_m^{(k)})^2 \geqslant 1 + (k-2)(u_2^{(k)})^2 \geqslant 1 + (k-2)k^2 \geqslant 10,$$

and the function $\log(2t)/t$ is decreasing for $t \geqslant 2$. So, we need a lower bound on $1 + (k-2)(u_m^{(k)})^2$.

It is well-known and easy to prove that if we write

$$\alpha_k = \frac{k + \sqrt{k^2-4}}{2} \qquad \text{and} \qquad \beta_k = \frac{k - \sqrt{k^2-4}}{2}$$

for the two roots of the quadratic equation $x^2 - kx + 1 = 0$, then

$$u_m^{(k)} = \frac{\alpha_k^m - \beta_k^m}{\alpha_k - \beta_k}.$$

Note that $\alpha_k - \beta_k = \sqrt{k^2-4}$ and $\alpha_k \beta_k = 1$. Hence,

$$\begin{aligned} 1 + (k-2)(u_m^{(k)})^2 &= 1 + \frac{k-2}{(\alpha_k - \beta_k)^2}\left(\alpha_k^{2m} + \beta_k^{2m} - 2\right) \\ &> 1 + \frac{1}{k+2}\left(\alpha_k^{2m} - 2\right) = \frac{\alpha_k^{2m} + k}{k+2} > \frac{\alpha_k^{2m}}{k+2} \\ &> \frac{(k^2-4)^m}{k+2} = (k-2)^m (k+2)^{m-1}. \end{aligned} \tag{2.3}$$

Note that for $k \geqslant 3$ and $m \geqslant 2$ we have that $(k-2)^m(k+2)^{m-1} \geqslant 5$. Thus,

$$\begin{aligned}\frac{1}{\phi(1+(k-2)(u_m^{(k)})^2)} &< \frac{\log(2(k-2)^m(k+2)^{m-1})}{(\log 2)(k-2)^m(k+2)^{m-1}} \\ &= \frac{1}{(k-2)^m(k+2)^{m-1}} \\ &+ \frac{m\log(k-2)}{(\log 2)(k-2)^m(k+2)^{m-1}} \\ &+ \frac{(m-1)\log(k+2)}{(\log 2)(k-2)^m(k+2)^{m-1}}.\end{aligned}$$

We shall apply the above inequality for all $k \geqslant 4$. The case $k = 3$ is special since in this case $u_m^{(2)} = F_{2m}$ for all $n \geqslant 1$, where $(F_m)_{m\geqslant 0}$ denotes the Fibonacci sequence given by $F_0 = 0,\ F_1 = 1$ and $F_{m+2} = F_{m+1} + F_m$ for all $m \geqslant 0$. Thus,

$$1 + (u_m^{(2)})^2 = 1 + F_{2m}^2 = F_{2m+1}F_{2m-1},$$

therefore

$$\phi(1+(u_m^{(2)})^2) = \phi(F_{2m+1}F_{2m-1}) = \phi(F_{2m+1})\phi(F_{2m-1}),$$

where the last relation holds because F_{2m+1} and F_{2m-1} are coprime. Summing up over all $m \geqslant 2$ and $k \geqslant 3$, we find that

$$S < S_0 + S_1 + S_2 + S_3,$$

where

$$\begin{aligned}S_0 &= \sum_{m\geqslant 2} \frac{1}{\phi(F_{2m+1})\phi(F_{2m-1})}, \\ S_1 &= \sum_{k\geqslant 4}\sum_{m\geqslant 2} \frac{1}{(k-2)^m(k+2)^{m-1}}, \\ S_2 &= \sum_{k\geqslant 4}\sum_{m\geqslant 2} \frac{m\log(k-2)}{(\log 2)(k-2)^m(k+2)^{m-1}}, \\ S_3 &= \sum_{k\geqslant 4}\sum_{m\geqslant 2} \frac{(m-1)\log(k+2)}{(\log 2)(k-2)^m(k+2)^{m-1}}.\end{aligned}$$

We now compute the four sums above. We computed,

$$S_0 < 0.277.$$

To deduce this inequality, we first computed the first 100 terms in S_0 getting an answer < 0.2769. For $n \geqslant 199$, we have $\phi(n) \geqslant 48$. Indeed, to see this note first that

$$\phi(n) \geqslant \frac{n\log 2}{\log(2n)} > 48,$$

where the inequality on the left holds always by inequality (2.2) and the inequality on the right holds for all $n \geqslant 500$. For $n \in [199, 500]$, we checked that the minimal value of the Euler function is 48. Next recall that a result of Luca [6] says that

$$\phi(F_n) \geqslant F_{\phi(n)}$$

holds for all n. In particular,

$$\frac{1}{\phi(F_{2n+1})\phi(F_{2n-1})} \leqslant \frac{1}{F_{\phi(2n+1)}F_{\phi(2n-1)}} \leqslant \frac{1}{\alpha^{\phi(2n+1)+\phi(2n-1)-4}},$$

where we use $\alpha = (1+\sqrt{5})/2$ together with the fact that the inequality $F_n \geqslant \alpha^{n-2}$ holds for all $n \geqslant 2$. Let $m = \phi(2n+1) + \phi(2n-1) - 4$. Since $n \geqslant 100$, we have that $2n - 1 \geqslant 199$, and so $m \geqslant 92$. Clearly,

$$4n - 4 > m > \frac{(2n+1)\log 2}{\log(4n+2)} + \frac{(2n-1)\log 2}{\log(4n-2)} - 4.$$

We checked that the square of the above lower bound is larger than the upper bound for all $n \geqslant 21$, which is our case. This implies that the number of n such that $\phi(2n+1) + \phi(2n-1) - 4 = m$ does not exceed m^2 for n in our range. Note that m is even. To summarize,

$$S_0 \leqslant \sum_{n=1}^{100} \frac{1}{\phi(F_{2n-1})\phi(F_{2n+1})} + \sum_{\ell \geqslant 46} \frac{4\ell^2}{\alpha^{2\ell}}.$$

For $\ell \geqslant 12$, we have that $\alpha^\ell \geqslant 4\ell^2$. Thus,

$$S_0 < \sum_{n=1}^{100} \frac{1}{\phi(F_{2n-1})\phi(F_{2n+1})} + \sum_{\ell \geqslant 46} \frac{1}{\alpha^\ell}$$

Thus, the error in approximating S_0 by its first 100 terms is

$$< \sum_{\ell \geqslant 46} \frac{1}{\alpha^\ell} = \frac{1}{\alpha^{45}(\alpha - 1)} < 10^{-9}.$$

So, indeed $S_0 < 0.277$. Next,

$$S_1 = \sum_{k \geqslant 4} \frac{1}{(k-2)} \sum_{m \geqslant 1} \frac{1}{(k^2-4)^m} = \sum_{k \geqslant 4} \frac{1}{(k-2)(k^2-5)} < 0.0861.$$

Further,

$$S_2 = \sum_{k \geqslant 4} \frac{\log(k-2)}{(\log 2)(k-2)} \sum_{m \geqslant 1} \frac{m+1}{(k^2-4)^m} < \sum_{k \geqslant 4} \frac{2(k+2)\log(k-2)}{(\log 2)(k^2-5)^2} < 0.2845.$$

Finally,

$$S_3 = \sum_{k\geqslant 4} \frac{\log(k+2)}{(\log 2)(k-2)} \sum_{m\geqslant 1} \frac{m}{(k^2-4)^m} = \sum_{k\geqslant 4} \frac{(k+2)\log(k+2)}{(\log 2)(k^2-5)^2} < 0.2607.$$

The upper bounds on S_1, S_2, S_3 were computed with Mathematica. We shall justify only S_1. Clearly,

$$\sum_{m\geqslant 1} \frac{1}{(k^2-4)^m} = \frac{1}{(k^2-4)} \cdot \frac{1}{1-\frac{1}{(k^2-4)}} = \frac{1}{(k^2-5)}.$$

With Mathematica, we obtained that

$$\sum_{k=4}^{1003} \frac{1}{(k-2)(k^2-5)} < 0.08607,$$

while certainly

$$\begin{aligned} \sum_{k>1003} \frac{1}{(k-2)(k^2-5)} &< \sum_{k>1003} \frac{1}{(k-2)^3} = \sum_{k>1001} \frac{1}{k^3} < \int_{1000}^{\infty} \frac{dt}{t^3} \\ &= -\frac{1}{2t^2}\Big|_{t=1000}^{t=\infty} = \frac{1}{2\cdot 10^6} < 0.00001, \end{aligned}$$

which together imply that $S_1 < 0.0861$, as claimed. A similar argument can be used to justify the bounds on S_2 and S_3. Hence,

$$S < 0.277 + 0.0861 + 0.2845 + 0.2607 = 0.9083 < 0.91,$$

which completes the proof of the lemma. □

Proof of Theorem 2.1. Assume that relation (1.1) does not hold with $m = 1$. Then we get that $(n-1)/(k-1)$ is an integer for some $3 \leqslant k < n$, and this certainly is the case for some k if $n-1$ is not a prime. From now on, we fix a large positive real number x and we look only at numbers $n \leqslant x$ such that $n-1$ is prime and relation (1.1) is not satisfied for some $3 \leqslant k < n$ and $m \geqslant 2$. Then

$$n - 1 \equiv k - 1 \pmod{1 + (k-2)(u_m^{(k)})^2}.$$

Since $k < n$, it follows that $n - 1 = (k-1) + \ell(1 + (k-2)(u_m^{(k)})^2)$ for some positive integer ℓ, therefore $1 + (k-2)(u_m^{(k)})^2 < x$. Since $m \geqslant 2$, it follows that

$$x > 1 + (k-2)(u_m^{(k)})^2 \geqslant (k-2)^m (k+2)^{m-1} \geqslant \max\{(k-2)^2(k+2), 5^{m-1}\}$$

(see estimate (2.3)), leading to $k = O(x^{1/3})$ and $m = O(\log x)$. So, there are only $O(x^{1/3}\log x)$ such pairs (k, m). We may further assume that $k-1$ is coprime to $1 + (k-2)(u_m^{(k)})^2$, for if not any common prime factor q of these two integers will

be $\leqslant k-1 < n-1$ and will divide $n-1$, which is impossible. For positive coprime integers a and b we write $\pi(x;a,b)$ for the number of primes $p \leqslant x$ which are congruent to $a \pmod b$ and we write $\pi(x)$ for the total number of prime numbers $p \leqslant x$. It then follows that the number of positive integers $n \leqslant x$ satisfying (1.1) for any $k \geqslant 3$ and $m \geqslant 1$ is

$$\#\mathcal{N}(x) \geqslant \pi(x-1) - \sum_{\substack{(k,m)\\ 1+(k-2)(u_m^{(k)})^2<x}} \pi(x; k-1, 1+(k-2)(u_m^{(k)})^2). \tag{2.4}$$

Thus, it suffices to show that the above expression exceeds $0.09x/\log x$ for all sufficiently large x.

Let x be large. We split the set of pairs (k,m) with $1+(k-2)(u_m^{(k)})^2 < x$ in three subsets as follows:

(i) $\mathcal{S}_1 = \{(k,m) : 1+(k-2)(u_m^{(k)})^2 < (\log x)^{10}\}$;

(ii) $\mathcal{S}_2 = \{(k,m) : (\log x)^{10} \leqslant 1+(k-2)(u_m^{(k)})^2 < x^{1/2}\}$;

(iii) $\mathcal{S}_3 = \{(k,m) : x^{1/2} \leqslant 1+(k-2)(u_m^{(k)})^2 < x\}$.

If $(k,m) \in \mathcal{S}_1$, then, by the Siegel-Walfiz theorem (see, for example, page 133 in [1]), we have that

$$\pi(x; k-1, 1+(k-2)(u_m^{(k)})^2) = \frac{\pi(x)}{\phi(1+(k-2)(u_m^{(k)})^2)} + O\left(\frac{x}{\exp(A\sqrt{\log x})}\right)$$

for some positive constant A. Note further that since for $(k,m) \in \mathcal{S}_1$ we have that

$$(\log x)^{10} \geqslant 1+(k-2)(u_m^{(k)})^2 \geqslant \max\{(k-2)^2(k+2), 5^{m-1}\},$$

we get $k \ll (\log x)^{10/3}$ and $m \ll \log\log x \ll (\log x)^{2/3}$, therefore

$$\#\mathcal{S}_1 \ll (\log x)^4.$$

If $(k,m) \in \mathcal{S}_2$, then by the Brun-Titchmarsh theorem (see, for example, [2, Section 2.3.1, Theorem 1] or [3, Chapter 3, Theorem 3.7]), we have that

$$\begin{aligned}\pi(x; k-1, 1+(k-2)(u_m^{(k)})^2) &\ll \frac{x}{\phi(1+(k-2)(u_m^{(k)})^2)\log\left(\frac{x}{1+(k-2)(u_m^{(k)})^2}\right)}\\ &\ll \frac{\pi(x)}{\phi(1+(k-2)(u_m^{(k)})^2)},\end{aligned}$$

where we used the fact that

$$\log\left(\frac{x}{1+(k-2)(u_m^{(k)})^2}\right) \geqslant \log(x^{1/2}) = \frac{\log x}{2},$$

as well as the Prime Number Theorem.

Finally, if $(k,m) \in \mathcal{S}_3$, then

$$\pi(x; k-1, 1+(k-2)(u_m^{(k)})^2) \leqslant \frac{x}{1+(k-2)(u_m^{(k)})^2} + 1 \ll x^{1/2}.$$

Putting everything together, we get that

$$\sum_{\substack{(k,m)\\ 1+(k-2)(u_m^{(k)})^2<x}} \pi(x; k-1, 1+(k-2)(u_m^{(k)})^2) \leqslant \pi(x) \sum_{(k,m)\in\mathcal{S}_1} \frac{1}{\phi(1+(k-2)(u_m^{(k)})^2)}$$
$$+ O\left(\frac{x(\log x)^4}{\exp(A\sqrt{\log x})} + \sum_{(k,m)\in\mathcal{S}_2} \frac{\pi(x)}{\phi(1+(k-2)(u_m^{(k)})^2)} + x^{1/2}\#\mathcal{S}_3 \right).$$

Note that $\#\mathcal{S}_3 \ll x^{1/3}\log x$, and by the Prime Number Theorem, we have

$$\frac{x(\log x)^4}{\exp(A\sqrt{\log x})} = o(\pi(x))$$

as $x \to \infty$. Since the series (2.1) sums to S, it follows that both estimates

$$\pi(x) \sum_{(k,m)\in\mathcal{S}_2} \frac{1}{\phi(1+(k-2)(u_m^{(k)})^2)} = o(\pi(x))$$
$$\pi(x) \sum_{(k,m)\in\mathcal{S}_1} \frac{1}{\phi(1+(k-2)(u_m^{(k)})^2)} = S\pi(x) + o(\pi(x))$$

hold as $x \to \infty$. Thus,

$$\sum_{\substack{(k,m)\\ 1+(k-2)(u_m^{(k)})^2<x}} \pi(x; k-1, 1+(k-2)(u_m^{(k)})^2) \leqslant \pi(x)(S+o(1)),$$

which together with estimate (2.4) and Lemma 2.2 implies the conclusion of the theorem. □

Acknowledgements. Work on this paper was done during a pleasant visit of F. L. at HRI in Allahabad, India. He thanks the people of that institute for their kind hospitality. During the preparation of this paper, F. L. was also supported in part by Grants SEP-CONACyT 46755 and PAPIIT IN 100508.

References

[1] Davenport, H., Multiplicative Number Theory, Second Edition, Grad. Text in Math. 74 *Springer-Verlag*, 1980.

[2] Greaves, G., Sieves in number theory, *Springer-Verlag*, Berlin, 2001.

[3] Halberstam, H., Richert, H.–E., Sieve methods, *Academic Press*, London, 1974.

[4] Herzberg, N.P., On a problem of Hurwitz, *Pacific J. Math.*, Vol. 50 (1974) 485–493.

[5] Hurwitz, A., Über eine Aufgabe der unbestimmten analysis, *Arch. Math. Phys.*, Vol. 3 (1907) 185–196.

[6] Luca, F., Euler indicators of Lucas sequences, *Bull. Mat. Soc. Mat. Roumaine*, Vol. 40 (88) (1997) 151–163.

Ioulia Baoulina
Chhatnag Road, Jhusi
Allahabad 211019
India

e-mail:
jbaulina@mail.ru
ioulia@hri.res.in

Florian Luca
C.P. 58089, Morelia
Michoacán
México

e-mail:
fluca@matmor.unam.mx

Annales Mathematicae et Informaticae
***35** (2008) pp. 21–30*
http://www.ektf.hu/ami

Connection between ordinary multinomials, Fibonacci numbers, Bell polynomials and discrete uniform distribution*

Hacène Belbachir, Sadek Bouroubi, Abdelkader Khelladi

Faculty of Mathematics, University of Sciences and Technology Houari Boumediene (U.S.T.H.B), Algiers, Algeria.

Submitted 8 July 2008; Accepted 16 September 2008

Abstract

Using an explicit computable expression of ordinary multinomials, we establish three remarkable connections, with the q-generalized Fibonacci sequence, the exponential partial Bell partition polynomials and the density of convolution powers of the discrete uniform distribution. Identities and various combinatorial relations are derived.

Keywords: Ordinary multinomials, Exponential partial Bell partition polynomials, Generalized Fibonacci sequence, Convolution powers of discrete uniform distribution.

MSC: 05A10, 11B39, 11B65, 60C05

1. Introduction

Ordinary multinomials are a natural extension of binomial coefficients, for an appropriate introduction of these numbers see Smith and Hogatt [18], Bollinger [6] and Andrews and Baxter [2]. These coefficients are defined as follows: Let $q \geqslant 1$ and $L \geqslant 0$ be integers. For an integer $a = 0, 1, \ldots, qL$, the ordinary multinomial $\binom{L}{a}_q$ is the coefficient of the a-th term of the following multinomial expansion

$$\left(1 + x + x^2 + \cdots + x^q\right)^L = \sum_{a \geqslant 0} \binom{L}{a}_q x^a, \tag{1.1}$$

with $\binom{L}{a}_1 = \binom{L}{a}$ (being the usual binomial coefficient) and $\binom{L}{a}_q = 0$ for $a > qL$.

*Research supported partially by LAID3 Laboratory of USTHB University.

Using the classical binomial coefficient, one has

$$\binom{L}{a}_q = \sum_{j_1+j_2+\cdots+j_q=a} \binom{L}{j_1}\binom{j_1}{j_2}\cdots\binom{j_{q-1}}{j_q}. \tag{1.2}$$

Some readily well known established properties are

the symmetry relation

$$\binom{L}{a}_q = \binom{L}{qL-a}_q, \tag{1.3}$$

the longitudinal recurrence relation

$$\binom{L}{a}_q = \sum_{m=0}^{q} \binom{L-1}{a-m}_q, \tag{1.4}$$

and the diagonal recurrence relation

$$\binom{L}{a}_q = \sum_{m=0}^{L} \binom{L}{m}\binom{m}{a-m}_{q-1}. \tag{1.5}$$

These coefficients, as for usual binomial coefficients, are built trough the Pascal triangle, known as "Generalized Pascal Triangle", see tables: 1, 2 and 3. One can find the first values of the generalized triangle in SLOANE [17] as A027907 for $q = 2$, A008287 for $q = 3$ and A035343 for $q = 4$.

As an illustration of recurrence relation, we give the triangles of trinomial, quadrinomial and pentanomial coefficients:

Table 1: Triangle of *trinomial* coefficients: $\binom{L}{a}_2$

$L\backslash a$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
0	1										
1	1	1	1								
2	1	2	3	2	1						
3	1	3	6	7	6	3	1				
4	1	4	10	16	19	16	10	4	1		
5	1	5	15	30	45	51	45	30	15	5	1

Table 2: Triangle of *quadrinomial* coefficients: $\binom{L}{a}_3$

$L\backslash a$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
0	1												
1	1	1	1	1									
2	1	2	3	4	3	2	1						
3	1	3	6	10	12	12	10	6	3	1			
4	1	4	10	20	31	40	44	40	31	20	10	4	1

Table 3: Triangle of *pentanomial* coefficients: $\binom{L}{a}_4$

$L\backslash a$	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	
0	1														
1	1	1	1	1	1										
2	1	2	3	4	5	4	3	2	1						
3	1	3	6	10	15	18	19	18	15	10	6	3	1		
4	1	4	10	20	35	52	68	80	85	80	68	52	35	20	$\cdots$

Several extensions and commentaries about these numbers have been investigated in the literature, for example Brondarenko [7] gives a combinatorial interpretation of ordinary multinomials $\binom{L}{a}_q$ as *the number of different ways of distributing* "a" *balls among* "L" *cells where each cell contains at most* "q" *balls.*

Using this combinatorial argument, one can easily establish the following relation

$$\begin{aligned}\binom{L}{a}_q &= \sum_{L_1+2L_2+\cdots+qL_q=a} \binom{L}{L_1}\binom{L-L_1}{L_2}\cdots\binom{L-L_1-\cdots-L_{q-1}}{L_q} \\ &= \sum_{L_1+2L_2+\cdots+qL_q=a} \binom{L}{L_1, L_2\cdots, L_q}. \end{aligned} \tag{1.6}$$

For a computational view of the relation (1.6) see Bollinger [6]. Andrews and Baxter [2] have considered the q-analog generalization of ordinary multinomials (see also [19] for an exhaustive bibliography). They have defined the q-multinomial coefficients as follows

$$\begin{bmatrix}L\\a\end{bmatrix}_q^{(p)} = \sum_{j_1+j_2+\cdots+j_q=a} q^{\sum_{l=1}^{q-1}(L-j_l)j_{l+1}-\sum_{l=q-p}^{q-1} j_{l+1}} \begin{bmatrix}L\\j_1\end{bmatrix}\begin{bmatrix}j_1\\j_2\end{bmatrix}\cdots\begin{bmatrix}j_{q-1}\\j_q\end{bmatrix}$$

where

$$\begin{bmatrix}L\\a\end{bmatrix} = \begin{bmatrix}L\\a\end{bmatrix}_q = \begin{cases} (q)_L \,/\, (q)_a\,(q)_{L-a} & \text{if } 0 \leqslant a \leqslant L \\ 0 & \text{otherwise}\end{cases}$$

is the usual q-binomial coefficient, and where $(q)_k = \prod_{m=1}^{\infty}(1-q^m)\,/\,(1-q^{k+m})$, is called q-series. This definition is motivated by the relation (1.2).

Another extension, the supernomials, has also been considered by Schilling and Warnaar [16]. These coefficients are defined to be the coefficients of x^a in the expression of $\prod_{j=1}^{N}\left(1+x+\cdots+x^j\right)^{L_j}$

A refinement of the q-multinomial coefficient is also considered for the trinomial case by Warnaar [20].

Barry [3] gives a generalized Pascal triangle as

$$\binom{n}{k}_{a(n)} := \prod_{j=1}^{k} a\,(n-j+1)\,/a\,(j)\,,$$

where $a(n)$ is a suitably chosen sequence of integers.

Kallas [11] and Noe [14] give a generalization of Pascal's triangle by considering the coefficient of x^a in the expression of $(a_0 + a_1 x + \cdots + a_q x^q)^L$.

The main goal of this paper is to give some connections of the ordinary multinomials with the generalized Fibonacci sequence, the exponential Bell polynomials, and the density of convolution powers of discrete uniform distribution. We will give also some interesting combinatorial identities.

2. A simple expression of ordinary multinomials

If we denote x_i the number of balls in a cell, the previous combinatorial interpretation given by Brondarenko is equivalent to evaluate the number of solutions of the system

$$\begin{cases} x_1 + \cdots + x_L = a, \\ 0 \leqslant x_1, \ldots, x_L \leqslant q. \end{cases} \tag{2.1}$$

Now, let us consider the system (2.1). For $t \in]-1, 1[$, we have (see also Comtet [8, Vol. 1, p. 92 (pb 16).])

$$\sum_{a \geqslant 0} \binom{L}{a}_q t^a = (1 + t + \cdots + t^q)^L = \sum_{0 \leqslant x_1, \ldots, x_L \leqslant q} t^{x_1 + \cdots + x_L},$$

and

$$\begin{aligned} (1 + t + \cdots + t^q)^L &= \left(1 - t^{q+1}\right)^L (1 - t)^{-L} \\ &= \left(\sum_{j=0}^{L} (-1)^j \binom{L}{j} t^{j(q+1)} \right) \left(\sum_{j \geqslant 0} \binom{j + L - 1}{L - 1} t^j \right). \end{aligned}$$

By identification, we obtain the following theorem.

Theorem 2.1. *The following identity holds*

$$\binom{L}{a}_q = \sum_{j=0}^{\lfloor a/(q+1) \rfloor} (-1)^j \binom{L}{j} \binom{a - j(q+1) + L - 1}{L - 1}. \tag{2.2}$$

This explicit relation seems to be important since in contrast to relations (1.2), (1.3) and (1.5), it allows to compute the ordinary multinomials with one summation symbol.

In 1711, de Moivre (see [13] or [12, 3rd ed. p. 39]) solves the system (2.1) as the right hand side of (2.2).

Corollary 2.2. *We have the following identity*

$$\sum_{j=0}^{\lfloor n/2 \rfloor} \binom{n}{j} \binom{n - j}{j} = \sum_{j=0}^{\lfloor n/3 \rfloor} (-1)^j \binom{n}{j} \binom{2n - 3j - 1}{n - 1}.$$

Proof. It suffices to use relation (6) in Theorem 2.1 for $q = 2$ and $a = L = n$. □

The left hand side of the equality has the following combinatorial meaning. It computes the number of ways to distribute n balls into n boxes with 2 balls at most into each box. Put a ball into each box, then choose j boxes for removing the boxes located in them into j boxes chosen from the remaining $n - j$ boxes.

3. Generalized Fibonacci sequences

Now, let us consider for $q \geqslant 1$, the "multibonacci" sequence $(\Phi_n^{(q)})_{n \geqslant -q}$ defined by

$$\begin{cases} \Phi_{-q}^{(q)} = \cdots = \Phi_{-2}^{(q)} = \Phi_{-1}^{(q)} = 0, \\ \Phi_0^{(q)} = 1, \\ \Phi_n^{(q)} = \Phi_{n-1}^{(q)} + \Phi_{n-2}^{(q)} + \cdots + \Phi_{n-q-1}^{(q)} \text{ for } n \geqslant 1. \end{cases}$$

In [4], Belbachir and Bencherif proved that

$$\Phi_n^{(q-1)} = \sum_{k_1+2k_2+\cdots+qk_q=n} \binom{k_1 + k_2 + \cdots + k_q}{k_1, k_2, \cdots, k_q},$$

and, for $n \geqslant 1$

$$\Phi_n^{(q-1)} = \sum_{k=0}^{\lfloor n/(q+1) \rfloor} (-1)^k \, \frac{n - k\,(q-1)}{n - kq} \binom{n - kq}{k} 2^{n-1-k(q+1)},$$

leading to

$$\sum_{k_1+\cdots+qk_q=n} \binom{k_1 + \cdots + k_q}{k_1, \cdots, k_q} = \sum_{k=0}^{\lfloor n/(q+1) \rfloor} (-1)^k \, \frac{n - k\,(q-1)}{n - kq} \binom{n - kq}{k} 2^{n-1-k(q+1)}.$$

This is an analogous situation in writing above a multiple summation with one symbol of summation. On the other hand, we establish a connection between the ordinary multinomials and the generalized Fibonacci sequence:

Theorem 3.1. *We have the following identity*

$$\Phi_n^{(q)} = \sum_{l=0}^{qm-r} \binom{n-l}{l}_q, \tag{3.1}$$

where m is given by the extended euclidean algorithm for division: $n = m\,(q+1) - r$, $0 \leqslant r \leqslant q$.

Proof. We have

$$\begin{aligned}
\Phi_n^{(q)} &= \sum_{k_1+2k_2+\cdots+(q+1)k_{q+1}=n} \binom{k_1+k_2+\cdots+k_{q+1}}{k_1,k_2,\cdots,k_{q+1}} \\
&= \sum_{L\geqslant 0} \sum_{k_1+2k_2+\cdots+(q+1)k_{q+1}=n} \binom{L}{k_1,k_2,\cdots,k_{q+1}} \\
&= \sum_{L\geqslant 0} \sum_{k_2+2k_3+\cdots+qk_{q+1}=n-L} \binom{L}{L-k_2-\cdots-k_{q+1},\ k_2,\cdots,\ k_{q+1}} \\
&= \sum_{L\geqslant 0} \binom{L}{n-L}_q \\
&= \sum_{L\geqslant \frac{n}{q+1}}^{n} \binom{L}{n-L}_q,
\end{aligned}$$

using the fact that $\binom{L}{a}_q = 0$ for $a<0$ or $a>qL$

Now consider the unique writing of n given by the extended euclidean algorithm for division: $n = m(q+1)-r$, $0\leqslant r<q+1$ then $\frac{n}{q+1} = m - \frac{r}{q+1}$, which gives

$$\Phi_n^{(q)} = \sum_{k=0}^{qm-r} \binom{m+k}{qm-r-k}_q = \sum_{k=0}^{qm-r} \binom{m+k}{(q+1)k+r}_q = \sum_{l=0}^{qm-r} \binom{n-l}{l}_q.$$

□

As an immediate consequence of Theorem 3.1, we obtain the following identities

$$\begin{aligned}
\Phi^{(q)}_{(q+1)m} &= \sum_{l=0}^{qm} \binom{(q+1)m-l}{l}_q = \sum_{k=0}^{qm} \binom{m+k}{(q+1)k}_q, \\
\Phi^{(q)}_{(q+1)m-1} &= \sum_{l=0}^{qm-1} \binom{(q+1)m-l-1}{l}_q = \sum_{k=0}^{qm} \binom{m+k}{(q+1)k+1}_q, \\
&\vdots \\
\Phi^{(q)}_{(q+1)m-r} &= \sum_{l=0}^{qm-r} \binom{(q+1)m-l-r}{l}_q = \sum_{k=0}^{qm} \binom{m+k}{(q+1)k+r}_q.
\end{aligned}$$

For $q=1$, we find the classical Fibonacci sequence:

$$F_{-1}=0,\ F_0=1,\ F_{n+1}=F_n+F_{n-1},\ \text{for } n\geqslant 0.$$

Thus, we obtain the well known identity

$$F_n = \sum_{l=0}^{\lfloor n/2\rfloor} \binom{n-l}{l}.$$

Recently, in [5], the first author and Szalay prove the unimodality of the sequence $u_k = \binom{n-k}{k}_q$ associated to generalized Fibonacci numbers. More generally, they establish the unimodality for all rays of generalized Pascal triangles by showing that the sequence $w_k = \binom{n+\alpha k}{m+\beta k}_q$ is log-concave, then unimodal.

4. Exponential partial Bell partition polynomials

In this section, we establish a connection of the ordinary multinomials with exponential partial Bell partition polynomials $B_{n,L}(t_1, t_2, \ldots)$ which are defined (see Comtet [8, p. 144]) as follows

$$\frac{1}{L!}\left(\sum_{m\geqslant 1}\frac{t_m}{m!}x^m\right)^L = \sum_{n\geqslant L} B_{n,L}\frac{x^n}{n!},\ L = 0, 1, 2, \ldots. \tag{4.1}$$

An exact expression of such polynomials is given by

$$B_{n,L}(t_1, t_2, \ldots) = \sum_{\substack{k_1+2k_2+\cdots=n\\ k_1+k_2+\cdots=L}} \frac{n!}{k_1!k_2!\cdots(1!)^{k_1}(2!)^{k_2}\cdots} t_1^{k_1}t_2^{k_2}\cdots.$$

In this expression, the number of variables is finite according to $k_1 + 2k_2 + \cdots = n$.

Next, we give some particular values of $B_{n,L}$:

$$\begin{aligned} B_{n,L}(1, 1, 1, \ldots) &= \left\{ n \atop L \right\} \text{ Stirling numbers of second kind,}\\ B_{n,L}(0!, 1!, 2!, \ldots) &= \left[n \atop L \right] \text{ Stirling numbers of first kind,}\\ B_{n,L}(1!, 2!, 3!, \ldots) &= \frac{n!}{L!}\binom{n-1}{n-L}. \end{aligned} \tag{4.2}$$

In [1], Abbas and Bouroubi give several extended values of $B_{n,L}$.

The connection with ordinary multinomials is given by the following result:

Theorem 4.1. *We have the following identity*

$$B_{n,L}(1!, 2!, \ldots, (q+1)!, 0, \ldots) = \frac{n!}{L!}\binom{L}{n-L}_q. \tag{4.3}$$

Proof. Taking in (4.1) $t_m = m!$ for $1 \leqslant m \leqslant q+1$ and zero otherwise, we obtain

$$\left(x + \cdots + x^{q+1}\right)^L = L!\sum_{n-L\geqslant 0} B_{n,L}(1!, 2!, \ldots, (q+1)!, 0, \ldots)\frac{x^n}{n!},$$

from which it follows

$$\sum_{a\geqslant 0}\binom{L}{a}_q x^a = \sum_{n-L\geqslant 0}\frac{L!}{n!}B_{n,L}\left(1!,2!,\ldots,(q+1)!,0,\ldots\right)x^{n-L}.$$

□

Corollary 4.2. *Let $q\geqslant 1$, $L\geqslant 0$ be integers, and $a\in\{0,1,\ldots,qL\}$. For $q\geqslant a$, we have the following identity*

$$\binom{L}{a}_q = \binom{L+a-1}{a}.$$

Proof. Using the fact that $B_{n,L}\left(1!,2!,\ldots,(q+1)!,0,\ldots\right) = B_{n,L}\left(1!,2!,3!,\ldots\right)$ for $q+1\geqslant n-L+1$, we obtain $\binom{L}{n-L}_q = \binom{n-1}{n-L}$ for $q\geqslant n-L$. We conclude with $a=n-L$. □

This is simply a combination with repetition permitted (i.e. multi combination).

5. Convolution powers of discrete uniform distribution

This section gives a connection between the ordinary multinomials and the convolution power of the discrete uniform distribution. The right hand side of identity (2.2) is a very well known expression. Indeed for $q,L\in\mathbb{N}$, let us denote by $U_q^{\star L}$ the L^{th} convolution power of the discrete uniform distribution

$$U_q := \frac{1}{q+1}\left(\delta_0+\delta_1+\cdots+\delta_q\right) \qquad (\delta_a \text{ is the Dirac measure}),$$

then for $a\in\mathbb{N}$ (see de Moivre [13] or [10]), with respect to the counting measure, its density is given by

$$P\left(U_q^{\star L}=a\right) = \frac{1}{(q+1)^L}\sum_{j=0}^{\lfloor a/(q+1)\rfloor}(-1)^j\binom{L}{j}\binom{a+L-(q+1)\,j-1}{L-1}. \tag{5.1}$$

Combining Theorem 2.1 and relation (5.1), we have the following result:

Corollary 5.1. *Using the above notations, we obtain the following identity*

$$P\left(U_q^{\star L}=a\right) = \frac{\binom{L}{a}_q}{(q+1)^L}.$$

It should be noted that the multinomials may be seen as the number of favorable cases to the realization of the elementary event $\{a\}$.

It is easy to show that the distribution of $U_q^{\star L}$ is symmetric by relation (1.3).

Corollary 5.2. *We have the following identities*

$$\begin{aligned}
\sum_{k=0}^{qL} k \binom{L}{k}_q &= (q+1)^L \frac{qL}{2}, \\
\sum_{k=0}^{qL} k^2 \binom{L}{k}_q &= (q+1)^L \frac{qL}{2}\left(\frac{qL}{2}+\frac{q+2}{6}\right), \\
\sum_{k=0}^{qL} k^3 \binom{L}{k}_q &= (q+1)^L \left(\frac{qL}{2}\right)^2\left(\frac{qL}{2}+\frac{q+2}{2}\right),
\end{aligned}$$

More generally, for $m \geqslant 1$, the following identity holds

$$\sum_{k=0}^{qL} k^m \binom{L}{k}_q = (q+1)^L \sum_{i_1+i_2+\cdots+i_L=m} \binom{m}{i_1, i_2, \ldots, i_L} u_{i_1} u_{i_2} \cdots u_{i_L},$$

where u_i is the i-th moment of the random variable U_q.

Proof. It suffices to compute the expectation of $U_q^{\star L}$ using, first the density distribution and second the summation of uniform distributions. It also comes from the application of the generating function of the distribution given by Corollary 5.1. □

Acknowledgements. The authors are grateful to Professor Miloud Mihoubi for pointing our attention to Bell polynomials. The authors are also grateful to the referee and would like to thank him/her for comments and suggestions which improved the quality of this paper.

References

[1] Abbas, M., Bouroubi, S., On new identities for Bell's polynomials, *Disc. Math.*, 293 (2005) 5–10.

[2] Andrews, G.E., Baxter, J., Lattice gas generalization of the hard hexagon model III q-trinomials coefficients, *J. Stat. Phys.*, 47 (1987) 297–330.

[3] Barry, P., On Integer-sequences-based constructions of generalized Pascal triangles. *Journal of integer sequences*, Vol. 9 (2006), Art. 06.2.4.

[4] Belbachir, H., Bencherif, F., Linear recurrent sequences and powers of a square matrix. *Integers*, 6 (2006), A12, 17 pp.

[5] Belbachir, H., Szalay, L., Unimodal rays in the regular and generalized Pascal triangles, *J. of Integer Seq.*, Vol. 11, Art. 08.2.4. (2008).

[6] Bollinger, R.C., A note on Pascal T-triangles, Multinomial coefficients and Pascal Pyramids, *The Fibonacci Quarterly*, 24 (1986) 140–144.

[7] Brondarenko, B.A., Generalized Pascal triangles and Pyramids, their fractals, graphs and applications, *The Fibonacci Association*, Santa Clara 1993, Translated from Russian by R.C. Bollinger.

[8] Comtet, L., Analyse combinatoire, *Puf, Coll. Sup. Paris*, (1970), Vol. 1 & Vol. 2.

[9] Graham, R.L., Knuth, D.E., Patashnik, O., Concrete mathematics, *Addison-Wesley*, 1994.

[10] Hald, A., A history of mathematical statistics from 1750 to 1930, *John Wiley, N. Y.*, 1998.

[11] Kallos, G., A generalization of Pascal triangles using powers of base numbers; *Annales mathématiques Blaise Pascal*, Vol. 13, no. 1 (2006) 1–15.

[12] de Moivre, A., The doctrine of chances, Third edition 1756 (first ed. 1718 and second ed. 1738), reprinted by *Chelsea, N. Y.*, 1967.

[13] de Moivre, A., Miscellanca Analytica de Scrichus et Quadraturis, *J. Tomson and J. Watts, London*, 1731.

[14] Noe, T.D., On the divisibility of generalized central trinomial coefficients, *Journal of Integer sequences*, Vol. 9 (2006) Art. 06.2.7.

[15] Philippou, A.N., A note of the Fibonacci sequence of order k and the Multinomial coefficients, *The Fibonacci Quarterly*, 21 (1983) 82–86.

[16] Schilling, A., Warnaar, S.O., Supernomial coefficients, Polynomial identities and q-series, *The Ramanujan J.*, 2 (1998) 459–494.

[17] Sloane, N.J.A., The online Encyclopedia of Integer sequences, Published electronically at `http://www.research.att.com/~njas/sequences`, 2008.

[18] Smith, C., Hogatt, V.E., Generating functions of central values of generalized Pascal triangles, *The Fibonacci Quarterly*, 17 (1979) 58–67.

[19] Warnaar, S.O., The Andrews-Gordon Identities and q-Multinomial coefficients, *Commun. Math. Phys*, 184 (1997) 203–232.

[20] Warnaar, S.O., Refined q-trinomial coefficients and character identities, *J. Statist. Phys.*, 102 (2001), no. 3–4, 1065–1081.

Hacène Belbachir
Sadek Bouroubi
Abdelkader Khelladi
USTHB, Faculté de Mathématiques
BP 32, El Alia
16111 Bab Ezzouar
Alger, Algérie

e-mail:
`hbelbachir@usthb.dz`, `hacenebelbachir@gmail.com`
`sbouroubi@usthb.dz`, `bouroubis@yahoo.fr`
`akhelladi@usthb.dz`, `khelladi@wissal.dz`

Annales Mathematicae et Informaticae
35 *(2008) pp. 31–42*
http://www.ektf.hu/ami

Quenching time of solutions for some nonlinear parabolic equations with Dirichlet boundary condition and a potential

Théodore K. Boni[a], Bernard Y. Diby[b]

[a]Institut National Polytechnique Houphouët-Boigny de Yamoussoukro
[b]Université d'Abobo-Adjamé, UFR-SFA
Département de Mathématiques et Informatiques

Submitted 9 June 2008; Accepted 15 September 2008

Abstract

In this paper, we address the following initial-boundary value problem

$$\begin{cases} u_t(x,t) = Lu(x,t) + r(x)(b-u(x,t))^{-p} & \text{in} \quad \Omega \times (0,T), \\ u(x,t) = 0 \quad \text{on} \quad \partial\Omega \times (0,T), \\ u(x,0) = u_0(x) \geqslant 0 \quad \text{in } \Omega, \end{cases}$$

where $p > 2$, Ω is a bounded domain in $\mathbb{R}^N$ with smooth boundary $\partial\Omega$, L is an elliptic operator, $b = \text{const} > 0$, $r \in C^1(\overline{\Omega})$, $\sup_{x\in\Omega} r(x) > 0$, $r(x)$ is nonnegative in Ω, $u_0 \in C^1(\overline{\Omega})$, $u_0(x)$ is a nonnegative in Ω, $\sup_{x\in\Omega} u_0(x) < b$. Under some assumptions, we show that the solution of the above problem quenches in a finite time, and its quenching time goes to that of the solution of the following differential equation $\alpha'(t) = r_0(b-\alpha(t))^{-p}$, $t > 0$, $\alpha(0) = M$, as M tends to b, where $M = \sup_{x\in\Omega} u_0(x)$ and $r_0 = \sup_{x\in\Omega} r(x)$. Finally, we give some numerical results to illustrate our analysis.

Keywords: Nonlinear parabolic equation, Dirichlet boundary condition, numerical quenching time, quenching

MSC: 35B40, 35B50, 35K60, 65M06

1. Introduction

Let Ω be a bounded domain in $\mathbb{R}^N$ with smooth boundary $\partial\Omega$. Consider the following initial-boundary value problem for a nonlinear parabolic equation with

Dirichlet boundary condition and a potential of the from

$$u_t(x,t) = Lu(x,t) + r(x)(b - u(x,t))^{-p} \quad \text{in} \quad \Omega \times (0,T), \tag{1.1}$$

$$u(x,t) = 0 \quad \text{on} \quad \partial\Omega \times (0,T), \tag{1.2}$$

$$u(x,0) = u_0(x) \geqslant 0 \quad \text{in} \quad \Omega, \tag{1.3}$$

where $p > 2$, $b = \text{const} > 0$,

$$Lu = \sum_{i,j=1}^{N} \frac{\partial}{\partial x_i}\left(a_{ij}(x)\frac{\partial u}{\partial x_j}\right),$$

where $a_{ij} : \overline{\Omega} \to \mathbb{R}$, $a_{ij} \in C^1(\overline{\Omega})$, $a_{ij} = a_{ji}$, $1 \leqslant i, j \leqslant N$, and there exists a constant $C > 0$ such that

$$\sum_{i,j=1}^{N} a_{ij}(x)\xi_i\xi_j \geqslant C\|\xi\|^2 \quad \forall x \in \overline{\Omega} \quad \forall \xi = (\xi_1, \ldots, \xi_N) \in \mathbb{R}^N,$$

where $\|.\|$ stands for the Euclidean norm of $\mathbb{R}^N$.

The initial data $u_0 \in C^1(\overline{\Omega})$, $u_0(x)$ is a nonnegative in Ω, $\sup_{x\in\Omega} u_0(x) < b$, $r \in C^1(\overline{\Omega})$, $r(x)$ is nonnegative in Ω, $\sup_{x\in\Omega} r(x) > 0$. Here, $(0,T)$ is the maximal time interval of existence of the solution u of (1.1)–(1.3), and by a solution, we mean the following.

Definition 1.1. A solution of (1.1)–(1.3) is a function $u(x,t)$ continuous in $\overline{\Omega} \times [0,T)$, $u(x,t) < b$ in $\overline{\Omega} \times [0,T)$, and twice continuously differentiable in x and once in t in $\Omega \times (0,T)$.

The time T may be finite or infinite. When T is infinite, then we say that the solution u exists globally. When T is finite, then the solution u develops a singularity in a finite time, namely,

$$\lim_{t\to T} \|u(\cdot,t)\|_\infty = b,$$

where $\|u(\cdot,t)\|_\infty = \max_{x\in\Omega} |u(x,t)|$. In this last case, we say that the solution u quenches in a finite time, and the time T is called the quenching time of the solution u.

Throughout this paper, we suppose that there exists $a \in \Omega$ such that

$$M = \sup_{x\in\Omega} u_0(x) = u_0(a) \quad \text{and} \quad r_0 = \sup_{x\in\Omega} r(x) = r(a).$$

Solutions of nonlinear parabolic equations which quench in a finite time have been the subject of investigations of many authors (see [3–5, 7, 9–15, 18, 21, 22, 24–26, 28, 29] and the references cited therein). In particular, the above problem has been studied by many authors, and by standard methods based on the maximum principle, local existence, uniqueness, quenching and global existence have been treated (see [7, 23, 24, 29]). In this paper, we are interested in the asymptotic

behavior of the quenching time. Our work was motivated by the paper of Friedman and Lacey in [16], where they have considered the following initial-boundary value problem

$$u_t = \epsilon \Delta u + f(u) \ \text{ in } \ \Omega \times (0,T),$$

$$u = 0 \ \text{ on } \ \partial\Omega \times (0,T),$$

$$u(x,0) = u_0(x) \geqslant 0 \ \text{ in } \ \Omega,$$

where $f(s)$ is positive, increasing, convex function for nonnegative values of s, $\int_0^\infty \frac{ds}{f(s)} < \infty$, ϵ is a positive parameter. The initial data $u_0(x)$ is a continuous function in Ω. Under some additional conditions on the initial data, they have proved that the solution u of the above problem blows up in a finite time, and its blow-up time goes to that of the solution of the following differential equation

$$\alpha'(t) = f(\alpha(t)), \quad \alpha(0) = M,$$

as ϵ goes to zero, where $M = \sup_{x\in\Omega} u_0(x)$ (we say that a solution blows up in a finite time if it attains the value infinity in a finite time). Also in [28], Nabongo and Boni have considered the problem (1.1)–(1.3) in the case where the potential $r(x) = 1$ and the operator L is replaced by ϵL. They have obtained a similar result as that found in [16] by Friedman and Lacey. Let us notice that for this kind of problems, other parameters have been taken such that the norm of the initial data (see, for instance [17]) in the case of blow-up problems. In the present paper, we also take the norm of the initial data as parameter and obtain an analogous result using both a modification of Kaplan's method (see [20]) and a method based on the construction of upper solutions. Our paper is written in the following manner. In the next section, under some conditions, we show that the solution u of (1.1)–(1.3) quenches in a finite time, and its quenching time goes to that of the solution of a certain differential equation as the norm of the initial data goes to b. Finally, in the last section, we give some numerical results to illustrate our analysis.

2. Quenching times

In this section, under some assumptions, we show that the solution u of (1.1)–(1.3) quenches in a finite time, and its quenching time tends to that of the solution of a certain differential equation as M tends to b.

In the introduction of the paper, we have mentioned that there exists $a \in \Omega$ such that $r_0 = \sup_{x\in\Omega} r(x) = r(a)$ and $M = \sup_{x\in\Omega} u_0(x) = u_0(a)$. Consider the following eigenvalue problem

$$-L\psi = \lambda_\delta \psi \ \text{ in } \ B(a,\delta), \tag{2.1}$$

$$\psi = 0 \ \text{ on } \ \partial B(a,\delta),$$

$$\psi > 0 \ \text{ in } \ B(a,\delta),$$

where $\delta > 0$, such that, $B(a,\delta) = \{x \in \mathbb{R}^N; \quad \|x-a\| < \delta\} \subset \Omega$. It is well known that the above eigenvalue problem admits a solution (ψ, λ_δ) such that $0 < \lambda_\delta \leqslant \frac{D}{\delta^2}$, where D is a positive constant which depends only on the upper bound of the coefficients of the operator L and the dimension N. We can normalize ψ so that $\int_{B(a,\delta)} \psi dx = 1$.

Now, we are in a position to state the main result of this paper.

Theorem 2.1. *Let K be an upper bound of the first derivatives of u_0 and r. Suppose that $\sup_{x\in\Omega} u_0(x) = M > 0$ and let $A = (1 + bDK^2 2^p)/r_0$. If*

$$b - M < \min\{1, A^{-3/(p+1)}, (K\mathrm{dist}(a, \partial\Omega))^{3/(p+1)}\},$$

then the solution u of (1.1)–(1.3) quenches in a finite time, and its quenching time T satisfies the following estimates

$$0 \leqslant T - T_M \leqslant \frac{1}{r_0}\left(1 + \frac{A}{p+1}\right)(b-M)^{(4p+1)/3} + o((b-M)^{(4p+1)/3}),$$

where $T_M = \frac{(b-M)^{p+1}}{r_0(p+1)}$ is the quenching time of the solution $\alpha(t)$ of the differential equation defined as follows

$$\alpha'(t) = r_0(b - \alpha(t))^{-p}, \quad t > 0, \quad \alpha(0) = M.$$

Proof. Since $u_0 \in C^1(\overline{\Omega})$ and $r \in C^1(\overline{\Omega})$, invoking the mean value theorem and the triangle inequality, we find that

$$u_0(x) \geqslant M - (b-M)^{(p+1)/3} \quad \text{for} \quad x \in B(a,\delta),$$

$$r(x) \geqslant r_0 - (b-M)^{(p+1)/3} \quad \text{for} \quad x \in B(a,\delta),$$

where $\delta = \frac{(b-M)^{(p+1)/3}}{K}$. Let $w(x,t)$ be the solution of the following initial-boundary value problem

$$w_t(x,t) - Lw(x,t) - r(x)(b - w(x,t))^{-p} = 0 \quad \text{in} \quad B(a,\delta)\times(0,T^*), \tag{2.2}$$

$$w(x,t) = 0 \quad \text{on} \quad \partial B(a,\delta)\times(0,T^*),$$

$$w(x,0) = u_0(x) \quad \text{in} \quad B(a,\delta),$$

where $(0,T^*)$ is the maximal time interval of existence of the solution w. By an application of the maximum principle, we see that w is nonnegative in $B(a,\delta)\times(0,T^*)$, because the initial data is nonnegative in $B(a,\delta)$. Introduce the function $v(t)$ defined as follows

$$v(t) = \int_{B(a,\delta)} w(x,t)\psi(x)dx \quad \text{for} \quad t \in [0,T^*).$$

Take the derivative of v in t and use (2.2) to obtain

$$v'(t) = \int_{B(a,\delta)} \psi Lw dx + \int_{B(a,\delta)} r(x)(b-w)^{-p}\psi dx \quad \text{for} \quad t \in (0,T^*).$$

Applying Green's formula, we arrive at

$$v'(t) = \int_{B(a,\delta)} wL\psi dx + \int_{B(a,\delta)} r(x)(b-w)^{-p}\psi dx \quad \text{for} \quad t \in (0, T^*).$$

Due to the fact that $r(x) \geqslant r_0 - (b-M)^{(p+1)/3} > 0$ for $x \in B(a,\delta)$, using (2.1) and Jensen's inequality, we discover that

$$v'(t) \geqslant -\lambda_\delta v(t) + (r_0 - (b-M)^{(p+1)/3})(b-v(t))^{-p}.$$

Let us notice that $0 \leqslant v(t) \leqslant b$ for $t \in (0, T^*)$, and

$$0 < \lambda_\delta \leqslant \frac{D}{\delta^2} = \frac{DK^2}{(b-M)^{(2p+2)/3}}.$$

We deduce that

$$v'(t) \geqslant r_0(b-v(t))^{-p}\left(1 - \frac{(b-M)^{(p+1)/3}}{r_0} - \frac{bDK^2(b-v(t))^p}{r_0(b-M)^{(2p+2)/3}}\right) \quad \text{for} \quad t \in (0, T^*).$$

Obviously, we have $(b-M)^{(p+1)/3} \leqslant (b-M)^{(p-2)/3}$ and

$$b - v(0) \leqslant b - M + (b-M)^{(p+1)/3} \leqslant 2(b-M),$$

which implies that

$$v'(0) \geqslant r_0(b-v(0))^{-p}(1 - A(b-M)^{(p-2)/3}) > 0.$$

We claim that

$$v'(t) > 0 \quad \text{for} \quad t \in (0, T^*).$$

To prove the claim, we argue by contradiction. Indeed, let t_0 be the first $t \in (0, T^*)$ such that $v'(t) > 0$ for $t \in [0, t_0)$ but $v'(t_0) = 0$. Thus, we have $v(t_0) \geqslant v(0)$, which implies that

$$0 = v'(t_0) \geqslant r_0(b-v(0))^{-p}(1 - A(b-M)^{(p-2)/3}) > 0.$$

But, this is a contradiction, and the claim is proved. Consequently, we get

$$b - v(t) \leqslant b - v(0) \leqslant 2(b-M) \quad \text{for} \quad t \in (0, T^*),$$

and with the help of the above inequalities, we arrive at

$$v'(t) \geqslant r_0(b-v(t))^{-p}(1 - A(b-M)^{(p-2)/3}) \quad \text{for} \quad t \in (0, T^*).$$

This estimate may be rewritten as follows

$$(b-v)^p dv \geqslant r_0(1 - A(b-M)^{(p-2)/3})\mathrm{d}t \quad \text{for} \quad t \in (0, T^*).$$

Integrate the above inequality over $(0, T^*)$ to obtain

$$\frac{(b - v(0))^{p+1}}{p+1} \geqslant r_0(1 - A(b-M)^{(p-2)/3})T^*,$$

which implies that

$$T^* \leqslant \frac{(b - M + (b-M)^{(p+1)/3})^{p+1}}{r_0(p+1)(1 - A(b-M)^{(p-2)/3})}.$$

We conclude that w quenches in a finite time because the quantity on the right hand side of the above inequality is finite. On the other hand, by the maximum principle, we have $u \geqslant 0$ in $\Omega \times (0, T)$. Exploiting this estimate, it is easy to see that

$$u_t - Lu - r(x)(1-u)^{-p} \geqslant w_t - Lw - r(x)(1-w)^{-p} \ \text{ in } \ B(a,\delta) \times (0, T_*),$$

$$u \geqslant w \ \text{ on } \ \partial B(a,\delta) \times (0, T_*),$$

$$u(x,0) \geqslant w(x,0) \ \text{ in } \ B(a,\delta),$$

where $T_* = \min\{T, T^*\}$. It follows from the maximum principle that

$$u(x,t) \geqslant w(x,t) \ \text{ in } \ B(a,\delta) \times (0, T_*),$$

which implies that

$$T \leqslant T^* \leqslant \frac{(b - M + (b-M)^{(p+1)/3})^{p+1}}{r_0(p+1)(1 - A(b-M)^{(p-2)/3})}. \tag{2.3}$$

Indeed, suppose that $T > T^*$. We have $\|u(\cdot, T^*)\|_\infty \geqslant \|w(\cdot, T^*)\|_\infty = b$. But, this is a contradiction because $(0, T)$ is the maximal time interval of existence of the solution u. Now, setting $z(x,t) = \alpha(t)$ in $\overline{\Omega} \times [0, T_0)$, it is not hard to see that

$$z_t - Lz - r(x)(1-z)^{-p} = 0 \ \text{ in } \ \Omega \times (0, T_0),$$

$$z \geqslant 0 \ \text{ on } \ \partial\Omega \times (0, T_0),$$

$$z(x,0) \geqslant u_0(x) \ \text{ in } \ \Omega.$$

The maximum principle implies that $0 \leqslant u(x,t) \leqslant z(x,t) = \alpha(t)$ in $\Omega \times (0, T^0)$, where $T^0 = \min\{T_0, T\}$. We infer that

$$T \geqslant T_0 = \frac{(b-M)^{p+1}}{r_0(p+1)}. \tag{2.4}$$

Indeed, suppose that $T_0 > T$, which implies that $\alpha(T) \geqslant \|u(\cdot, T)\|_\infty = b$. But, this is a contradiction because $(0, T_0)$ is the maximal time interval of existence of the solution $\alpha(t)$. Apply Taylor's expansion to obtain

$$(b - M + (b-M)^{(p+1)/3})^{p+1} = (b-M)^{p+1}$$

$$+(p+1)(b-M)^{(4p+1)/3}+o((b-M)^{(4p+1)/3}),$$

$$\frac{1}{1-A(b-M)^{(p-2)/3}}=1+A(b-M)^{(p-2)/3}+o((b-M)^{(p-2)/3}).$$

Use (2.3), (2.4) and the above relations to complete the rest of the proof. □

Remark 2.2. Let us notice that the estimates obtained in Theorem 2.1 may be rewritten in the following form

$$0\leqslant\frac{T}{T_M}-1\leqslant(p+1+A)(b-M)^{(p-2)/3}+o((b-M)^{(p-2)/3}).$$

We deduce that $\lim_{M\to b}\frac{T}{T_M}=1$.

3. Numerical results

In this section, we give some computational results to confirm the theory established in the previous section. We consider the radial symmetric solution of the initial-boundary value problem below

$$u_t=\Delta u+\frac{1}{\|x\|+1}(1-u)^{-p}\ \text{ in }\ B\times(0,T),$$

$$u=0\ \text{ on }\ S\times(0,T),$$

$$u(x,0)=u_0(x)\ \text{ in }\ B,$$

where $B=\{x\in\mathbb{R}^N;\|x\|<1\}$, $S=\{x\in\mathbb{R}^N;\|x\|=1\}$ and $u_0(x)=M\cos(\frac{\pi\|x\|}{2})$ with $M\in(0,1)$. The above problem may be rewritten in the following form

$$u_t=u_{rr}+\frac{N-1}{r}u_r+\frac{1}{r+1}(1-u)^{-p},\quad r\in(0,1),\quad t\in(0,T),\tag{3.1}$$

$$u_r(0,t)=0,\quad u(1,t)=0,\quad t\in(0,T),\tag{3.2}$$

$$u(r,0)=\varphi(r),\quad r\in(0,1),\tag{3.3}$$

where $\varphi(r)=M\cos(\frac{\pi r}{2})$. We start by the construction of some adaptive schemes as follows. Let I be a positive integer and let $h=1/I$. Define the grid $x_i=ih$, $0\leqslant i\leqslant I$, and approximate the solution u of (3.1)–(3.3) by the solution $U_h^{(n)}=(U_0^{(n)},\dots,U_I^{(n)})^T$ of the following explicit scheme

$$\frac{U_0^{(n+1)}-U_0^{(n)}}{\Delta t_n}=N\frac{2U_1^{(n)}-2U_0^{(n)}}{h^2}+(1-U_0^{(n)})^{-p},$$

$$\frac{U_i^{(n+1)}-U_i^{(n)}}{\Delta t_n}=\frac{U_{i+1}^{(n)}-2U_i^{(n)}+U_{i-1}^{(n)}}{h^2}+\frac{(N-1)}{ih}\frac{U_{i+1}^{(n)}-U_{i-1}^{(n)}}{2h}$$

$$+\frac{1}{ih+1}(1-U_i^{(n)})^{-p}, \quad 1 \leqslant i \leqslant I-1,$$

$$U_I^{(n)} = 0, \quad U_i^{(0)} = M\cos\left(\frac{ih\pi}{2}\right), \quad 0 \leqslant i \leqslant I,$$

where $n \geqslant 0$. In order to permit the discrete solution to reproduce the properties of the continuous one when the time t approaches the quenching time T, we need to adapt the size of the time step so that we take

$$\Delta t_n = \min\left\{\frac{h^2}{2N}, h^2(1-\|U_h^{(n)}\|_\infty)^{p+1}\right\}$$

with $\|U_h^{(n)}\|_\infty = \sup_{0\leqslant i\leqslant I}|U_i^{(n)}|$. Let us notice that the restriction on the time step ensures the nonnegativity of the discrete solution. We also approximate the solution u of (3.1)–(3.3) by the solution $U_h^{(n)}$ of the implicit scheme below

$$\frac{U_0^{(n+1)}-U_0^{(n)}}{\Delta t_n} = N\frac{2U_1^{(n+1)}-2U_0^{(n+1)}}{h^2} + (1-U_0^{(n)})^{-p},$$

$$\frac{U_i^{(n+1)}-U_i^{(n)}}{\Delta t_n} = \frac{U_{i+1}^{(n+1)}-2U_i^{(n+1)}+U_{i-1}^{(n+1)}}{h^2} + \frac{(N-1)}{ih}\frac{U_{i+1}^{(n+1)}-U_{i-1}^{(n+1)}}{2h}$$
$$+\frac{1}{ih+1}(1-U_i^{(n)})^{-p}, \quad 1 \leqslant i \leqslant I-1,$$

$$U_I^{(n+1)} = 0, \quad U_i^{(0)} = M\cos\left(\frac{ih\pi}{2}\right), \quad 0 \leqslant i \leqslant I.$$

As in the case of the explicit scheme, here, we also choose

$$\Delta t_n = h^2(1-\|U_h^{(n)}\|_\infty)^{p+1}.$$

For the above implicit scheme, the existence and nonnegativity of the discrete solution are also guaranteed using standard methods (see, for instance [6]).
We note that

$$\lim_{r\to 0}\frac{u_r(r,t)}{r} = u_{rr}(0,t),$$

which implies that

$$u_t(0,t) = Nu_{rr}(0,t) + (1-u(0,t))^{-p} \quad \text{for} \quad t \in (0,T).$$

This observation has been taken into account in the construction of the above schemes at the first node. We need the following definition.

Definition 3.1. We say that the discrete solution $U_h^{(n)}$ of the explicit scheme or the implicit scheme quenches in a finite time if $\lim_{n\to\infty}\|U_h^{(n)}\|_\infty = 1$, and the series $\sum_{n=0}^{\infty}\Delta t_n$ converges. The quantity $\sum_{n=0}^{\infty}\Delta t_n$ is called the numerical quenching time of the discrete solution $U_h^{(n)}$.

In the following tables, in rows, we present the numerical quenching times, the numbers of iterations, the CPU times and the orders of the approximations corresponding to meshes of 16, 32, 64, 128. We take for the numerical quenching time $t_n = \sum_{j=0}^{n-1} \Delta t_j$ which is computed at the first time when

$$\Delta t_n = |t_{n+1} - t_n| \leqslant 10^{-16}.$$

The order (s) of the method is computed from

$$s = \frac{\log((T_{4h} - T_{2h})/(T_{2h} - T_h))}{\log 2}.$$

Numerical experiments

First case: $p = 3$, $N = 2$, $M = 0.90$

Table 1. Numerical quenching times, numbers of iterations, CPU times (seconds) and orders of the approximations obtained with the explicit Euler method.

I	t_n	n	CPU_t	s
16	2.5257 e-5	1361	1	-
32	2.5174 e-5	5100	3	-
64	2.5186 e-5	19007	32	2.79
128	2.5226 e-5	70461	2182	1.74

Table 2. Numerical quenching times, numbers of iterations, CPU times (seconds) and orders of the approximations obtained with the implicit Euler method.

I	t_n	n	CPU_t	s
16	2.5258 e-5	1361	1	-
32	2.5174 e-5	5100	6	-
64	2.5186 e-5	19007	155	2.81
128	2.5226 e-5	70461	5534	1.74

Second case: $p = 3$, $N = 2$, $M = 0.95$

Table 3. Numerical quenching times, numbers of iterations, CPU times (seconds) and orders of the approximations obtained with the explicit Euler method.

I	t_n	n	CPU_t	s
16	1.5725 e-6	1183	1	-
32	1.5657 e-6	4384	3	-
64	1.5642 e-6	16124	44	2.18
128	1.5641 e-6	58833	2373	3.91

Table 4. Numerical quenching times, numbers of iterations, CPU times (seconds) and orders of the approximations obtained with the implicit Euler method.

I	t_n	n	CPU_t	s
16	1.5725 e-6	1183	1	-
32	1.5657 e-6	4384	4	-
64	1.5642 e-6	16124	103	2.18
128	1.5641 e-6	58833	3366	3.91

Remark 3.2. If we consider the problem (3.1)–(3.3) in the case where the initial data $\varphi(r) = 0.9\cos(\frac{\pi r}{2})$ and $p = 3$, then it is not hard to see that the quenching time of the solution of the differential equation defined in Theorem 2.1 equals 2.5 e-5. We observe from Tables 1-2 that the numerical quenching time is approximately equal 2.5 e-5. This result has been proved in Theorem 2.1. When the initial data $\varphi(r) = 0.95\cos(\frac{\pi r}{2})$ and $p = 3$, then we find that the quenching time of the solution of the differential equation defined in Theorem 2.1 equals 1.5625 e-5. We discover from Tables 3–4 that the numerical quenching time is approximately equal 1.5625 e-6 which is a result proved in Theorem 2.1.

Acknowledgements. The authors want to thank the anonymous referee for the throughout reading of the manuscript and valuable comment that help us improve the presentation of the paper.

References

[1] Abia, L.M., López-Marcos, J.C., Martínez, J., On the blow-up time convergence of semidiscretizations of reaction-diffusion equations, *Appl. Numer. Math.*, 26 (1998) 399–414.

[2] Abia, L.M., López-Marcos, J.C. Martínez, J., Blow-up for semidiscretizations of reaction-diffusion equations, *Appl. Numer. Math.*, 20 (1996) 145–156.

[3] Acker, A., Walter, W., The quenching problem for nonlinear parabolic differential equations, *Lecture Notes in Math., Springer-Verlag, New York*, 564 (1976) 1–12.

[4] Acker, A., Kawohl, B., Remarks on quenching, *Nonl. Anal. TMA*, 13 (1989) 53–61.

[5] Bandle, C., Braumer, C.M., Singular perturbation method in a parabolic problem with free boundary, *in Proc. BAIL IVth Conference, Boole Press Conf. Ser. 8, Novosibirsk.*, (1986) 7–14.

[6] Boni, T.K., Extinction for discretizations of some semilinear parabolic equations, *C. R. Acad. Sci. Paris*, Serie I 333 (2001) 795–800.

[7] Boni, T.K., On quenching of solutions for some semilinear parabolic equations of second order, *Bull. Belg. Math. Soc.*, 7 (2000) 73–95.

[8] Brezis, H., Cazenave, T., Martel, Y., Ramiandrisoa, A., Blow-up of $u_t = u_{xx} + g(u)$ revisited, *Adv. Diff. Eq.*, 1 (1996) 73–90.

[9] Chan, C.Y., Lan Ke, Beyond quenching for singular reaction-diffusion problem, *Mathematical Methods in the Applied Sciences*, 17 (1994) 1–9.

[10] Chan, C.Y., New results in quenching, *in Proc. 1st World Congress Nonlinear Anal.*, (1996) 427–434.

[11] Chan, C.Y., Recent advances in quenching phenomena, *Pro. Dynamic Systems and Appl.*, 2 (1996) 107–113.

[12] Deng, K., Levine, H.A., On the blow-up of u_t at quenching, *Proc. Amer. Math. Soc.*, 106 (1989) 1049–1056.

[13] Deng, K., Xu, M., Quenching for a nonlinear diffusion equation with singular boundary condition, *Z. angew. Math. Phys.*, 50 (1999) 574–584.

[14] Fila, M., Kawohl, B., Levine, H.A., Quenching for quasilinear equations, *Comm. Part. Diff. Equat.*, 17 (1992) 593–614.

[15] Fila, M., Levine, H.A., Quenching on the boundary, *Nonl. Anal. TMA*, 21 (1993) 795–802.

[16] Friedman, A., Lacey, A.A., The blow-up time for solutions of nonlinear heat equations with small diffusion, *SIAM J. Math. Anal.*, 18 (1987) 711–721.

[17] Gui, G., Wang, X., Life span of solutions of the Cauchy problem for a nonlinear heat equation, *J. Diff. Equat.*, 115 (1995) 162–172.

[18] Guo, J., On a quenching problem with Robin boundary condition,*Nonl. Anal. TMA*, 17 (1991) 803–809.

[19] Ishige, K., Yagisita, H., Blow-up problems for a semilinear heat equation with large diffusion, *J. Diff. Equat.*, 212 (2005) 114–128.

[20] Kaplan, S., On the growth of the solutions of quasi-linear parabolic equation, *Comm. Appl. Math. Anal.*, 16 (1963) 305–330.

[21] Kawarada, H., On solutions of initial-boundary problem for $u_t = u_{xx} + 1/(1-u)$, *Pul. Res. Inst. Math. Sci.*, 10 (1975) 729–736.

[22] Kirk, C.M., Roberts, C.A., A review of quenching results in the context of nonlinear Volterra equations, *Dyn. Contin. Discrete Impuls. Syst. Ser. A Math. Anal.*, 10 (2003) 343–356.

[23] Ladyzenskaya, O.A., Solonnikov, V.A., Ural' Ceva, N.N., Linear and quasilinear equations of parabolic type, *Trans. Math. Monogr.*, 23 AMS, Providence, RI, (1968).

[24] Levine, H.A., The phenomenon of quenching: a survey, in Trends In The Theory And Practice Of Nonlinear Analysis, *North-Holland, Amsterdam*, (1985) 275–286.

[25] Levine, H.A., The quenching of solutions of linear parabolic and hyperbolic equations with nonlinear boundary conditions, *SIAM J. Math. Anal.*, 14 (1983) 1139–1152.

[26] Levine, H.A., Quenching, nonquenching and beyond quenching for solution of some parabolic equations, *Ann. Math. Pura Appl.*, 155 (1989) 243–260.

[27] Nakagawa, T., Blowing up on the finite difference solution to $u_t = u_{xx} + u^2$, *Appl. Math. Optim.*, 2 (1976) 337–350.

[28] Nabongo, D., Boni, T.K., Quenching time for some nonlinear parabolic equations, *An. St. Univ. Ovidius Constanta*, 16 (2008) 91–106.

[29] Phillips, D., Existence of solution of quenching problems, *Appl. Anal.*, 24 (1987) 253–264.

[30] Protter, M.H., Weinberger, H.F., Maximum principles in differential equations, *Prentice Hall, Englewood Cliffs, NJ*, (1967).

[31] Shang, Q., Khaliq, A.Q.M., A compound adaptive approach to degenerate nonlinear quenching problems, *Numer. Meth. Part. Diff. Equat.*, 15 (1999) 29–47.

[32] Walter, W., Differential-und Integral-Ungleichungen, *Springer, Berlin*, (1964).

Théodore K. Boni
Institut National Polytechnique Houphouët-Boigny de Yamoussoukro
BP 1093 Yamoussoukro, (Côte d'Ivoire)
e-mail: theokboni@yahoo.fr

Bernard Y. Diby
Université d'Abobo-Adjamé, UFR-SFA
Département de Mathématiques et Informatiques
02 BP 801 Abidjan 02, (Côte d'Ivoire)
e-mail: ydiby@yahoo.fr

Annales Mathematicae et Informaticae
***35** (2008) pp. 43–59*
http://www.ektf.hu/ami

Common fixed point theorems for pairs of single and multivalued D-maps satisfying an integral type

H. Bouhadjera, A. Djoudi

Laboratoire de Mathématiques Appliquées
Université Badji Mokhtar, Annaba, Algérie

Submitted 4 March 2008; Accepted 28 June 2008

Abstract

This contribution is a continuation of [1, 3, 14]. The concept of subcompatibility between single maps and between single and multivalued maps is used as a tool for proving existence and uniqueness of common fixed points on complete metric and symmetric spaces. Extensions of known results, in particularly results given by Djoudi and Aliouche, Elamrani and Mehdaoui, Pathak et al. are thereby obtained.

Keywords: Commuting and weakly commuting maps, compatible and compatible maps of type (A), (B), (C) and (P), weakly compatible maps, δ-compatible maps, subcompatible maps, D-maps, integral type, common fixed point theorems, metric space.

MSC: 47H10, 54H25

1. Introduction and preliminaries

Let $(\mathcal{X}, d)$ be a metric space and let $B(\mathcal{X})$ be the class of all nonempty bounded subsets of $\mathcal{X}$. For all A, B in $B(\mathcal{X})$, define

$$\delta(A, B) = \sup \{d(a, b) : a \in A, b \in B\}.$$

If $A = \{a\}$, we write $\delta(A, B) = \delta(a, B)$. Also, if $B = \{b\}$, it yields that $\delta(A, B) = d(a, b)$.

From the definition of $\delta(A, B)$, for all A, B, C in $B(\mathcal{X})$ it follows that

$$\delta(A, B) = \delta(B, A) \geqslant 0,$$
$$\delta(A, B) \leqslant \delta(A, C) + \delta(C, B),$$

$$\delta(A, A) = \text{diam}A,$$
$$\delta(A, B) = 0 \quad \text{iff} \quad A = B = \{a\}.$$

In his paper [15], Sessa introduced the notion of weak commutativity which generalized the notion of commutativity.

Later on, Jungck [6] gave a generalization of weak commutativity by introducing the concept of compatibility.

Again, to generalize weakly commuting maps, the same author with Murthy and Cho [8] introduced the concept of compatible maps of type (A).

Extending type (A), Pathak and Khan [13] made the notion of compatible maps of type (B).

In [11], the concept of compatible maps of type (P) was introduced and compared with compatible and compatible maps of type (A).

In 1998, Pathak, Cho, Kang and Madharia [12] defined the notion of compatible maps of type (C) as another extension of compatible maps of type (A).

In his paper [7], Jungck generalized all the concepts of compatibility by giving the notion of weak compatibility (subcompatibility).

The authors of [9] extended the concept of compatible maps to the setting of single and multivalued maps by giving the notion of δ-compatible maps.

Also, the same authors [10] extended the definition of weak compatibility to the setting of single and multivalued maps by introducing the concept of subcompatible maps.

In their paper [2], Djoudi and Khemis introduced the notion of D-maps which is a generalization of δ-compatible maps.

Definition 1.1 ([4]). A sequence $\{A_n\}$ of nonempty subsets of $\mathcal{X}$ is said to be convergent to a subset A of $\mathcal{X}$ if:

(i) each point $a \in A$ is the limit of a convergent sequence $\{a_n\}$, where $a_n \in A_n$ for $n \in \mathbb{N}$,

(ii) for arbitrary $\epsilon > 0$, there exists an integer m such that $A_n \subseteq A_\epsilon$ for $n > m$, where A_ϵ denotes the set of all points x in $\mathcal{X}$ for which there exists a point a in A, depending on x, such that $d(x, a) < \epsilon$.

Lemma 1.2 ([4, 5]). *If $\{A_n\}$ and $\{B_n\}$ are sequences in $B(\mathcal{X})$ converging to A and B in $B(\mathcal{X})$, respectively, then the sequence $\{\delta(A_n, B_n)\}$ converges to $\delta(A, B)$.*

Lemma 1.3 ([5]). *Let $\{A_n\}$ be a sequence in $B(\mathcal{X})$ and y be a point in $\mathcal{X}$ such that $\delta(A_n, y) \to 0$. Then the sequence $\{A_n\}$ converges to the set $\{y\}$ in $B(\mathcal{X})$.*

Definition 1.4 ([15]). The self-maps f and g of a metric space $\mathcal{X}$ are said to be weakly commuting if $d(fgx, gfx) \leqslant d(gx, fx)$ for all $x \in \mathcal{X}$.

Definition 1.5 ([6, 8, 13, 12, 11]). The self-maps f and g of a metric space $\mathcal{X}$ are said to be

(1) compatible if

$$\lim_{n\to\infty} d(fgx_n, gfx_n) = 0,$$

(2) compatible of type (A) if

$$\lim_{n\to\infty} d(fgx_n, g^2x_n) = 0 \text{ and } \lim_{n\to\infty} d(gfx_n, f^2x_n) = 0,$$

(3) compatible of type (B) if

$$\lim_{n\to\infty} d(fgx_n, g^2x_n) \leqslant \frac{1}{2}\left[\lim_{n\to\infty} d(fgx_n, ft) + \lim_{n\to\infty} d(ft, f^2x_n)\right],$$
$$\lim_{n\to\infty} d(gfx_n, f^2x_n) \leqslant \frac{1}{2}\left[\lim_{n\to\infty} d(gfx_n, gt) + \lim_{n\to\infty} d(gt, g^2x_n)\right],$$

(4) compatible of type (C) if

$$\lim_{n\to\infty} d(fgx_n, g^2x_n) \leqslant \frac{1}{3}\Big[\lim_{n\to\infty} d(fgx_n, ft) + \lim_{n\to\infty} d(ft, f^2x_n) + \lim_{n\to\infty} d(ft, g^2x_n)\Big],$$
$$\lim_{n\to\infty} d(gfx_n, f^2x_n) \leqslant \frac{1}{3}\Big[\lim_{n\to\infty} d(gfx_n, gt) + \lim_{n\to\infty} d(gt, g^2x_n) + \lim_{n\to\infty} d(gt, f^2x_n)\Big],$$

(5) compatible of type (P) if

$$\lim_{n\to\infty} d(f^2x_n, g^2x_n) = 0$$

whenever $\{x_n\}$ is a sequence in $\mathcal{X}$ such that $\lim_{n\to\infty} fx_n = \lim_{n\to\infty} gx_n = t$ for some $t \in \mathcal{X}$.

Definition 1.6 ([7]). The self-maps f and g of a metric space $\mathcal{X}$ are called weakly compatible if $fx = gx$, $x \in \mathcal{X}$ implies $fgx = gfx$.

Definition 1.7 ([9]). The maps $f\colon \mathcal{X} \to \mathcal{X}$ and $F\colon \mathcal{X} \to B(\mathcal{X})$ are δ-compatible if

$$\lim_{n\to\infty} \delta(Ffx_n, fFx_n) = 0$$

whenever $\{x_n\}$ is a sequence in $\mathcal{X}$ such that $fFx_n \in B(\mathcal{X})$, $fx_n \to t$ and $Fx_n \to \{t\}$ for some $t \in \mathcal{X}$.

Definition 1.8 ([10]). Maps $f\colon \mathcal{X} \to \mathcal{X}$ and $F\colon \mathcal{X} \to B(\mathcal{X})$ are subcompatible if they commute at coincidence points; i.e., for each point $u \in \mathcal{X}$ such that $Fu = \{fu\}$, we have $Ffu = fFu$.

Definition 1.9 ([2]). The maps $f\colon \mathcal{X} \to \mathcal{X}$ and $F\colon \mathcal{X} \to B(\mathcal{X})$ are said to be D-maps iff there exists a sequence $\{x_n\}$ in $\mathcal{X}$ such that for some $t \in \mathcal{X}$

$$\lim_{n\to\infty} fx_n = t \text{ and } \lim_{n\to\infty} Fx_n = \{t\}.$$

Recently in 2007, Pathak et al. [14] established a general common fixed point theorem for two pairs of weakly compatible maps satisfying integral type implicit relations. The first main object of this paper is to prove a common fixed point theorem for a quadruple of maps satisfying certain integral type implicit relations. Our result extended the result of [14] to the setting of single and multivalued maps.

For this consideration we need the following:

Let $\Phi = \{\varphi : \mathbb{R}_+ \to \mathbb{R}$ is a Lebesgue-integrable map which is summable$\}$ and let $\mathcal{F}$ be the set of all continuous functions $F\colon \mathbb{R}_+^6 \to \mathbb{R}_+$ satisfying the following conditions:

(F_a) $\int_0^{F(u,0,0,u,u,0)} \varphi(t)dt \leqslant 0$ implies $u = 0$;

(F_b) $\int_0^{F(u,0,u,0,0,u)} \varphi(t)dt \leqslant 0$ implies $u = 0$.

The function F satisfies the condition (F_1) if $\int_0^{F(u,u,0,0,u,u)} \varphi(t)dt > 0$ for all $u > 0$.

2. Main results

Theorem 2.1. *Let f, g be self-maps of a metric space $(\mathcal{X}, d)$ and let $F, G\colon \mathcal{X} \to B(\mathcal{X})$ be two multivalued maps such that*

(1) *$F\mathcal{X} \subseteq g\mathcal{X}$ and $G\mathcal{X} \subseteq f\mathcal{X}$,*

(2)

$$\int_0^{F(\delta(Fx,Gy),d(fx,gy),\delta(fx,Fx),\delta(gy,Gy),\delta(fx,Gy),\delta(gy,Fx))} \varphi(t)\,dt \leqslant 0$$

for all x, y in $\mathcal{X}$, where $F \in \mathcal{F}$ and $\varphi \in \Phi$. If either

(3) *f and F are subcompatible D-maps; g and G are subcompatible and $F\mathcal{X}$ is closed, or*

(3′) *g and G are subcompatible D-maps; f and F are subcompatible and $G\mathcal{X}$ is closed.*

Then, f, g, F and G have a unique common fixed point $t \in \mathcal{X}$ such that

$$Ft = Gt = \{ft\} = \{gt\} = \{t\}.$$

Proof. Suppose that f and F are D-maps, then, there exists a sequence $\{x_n\}$ in $\mathcal{X}$ such that $fx_n \to t$ and $Fx_n \to \{t\}$ for some $t \in \mathcal{X}$. Since $F\mathcal{X}$ is closed and $F\mathcal{X} \subseteq g\mathcal{X}$, then, there is a point $u \in \mathcal{X}$ such that $gu = t$. We show that $Gu = \{gu\} = \{t\}$. Using inequality (2), we have

$$\int_0^{F(\delta(Fx_n,Gu),d(fx_n,gu),\delta(fx_n,Fx_n),\delta(gu,Gu),\delta(fx_n,Gu),\delta(gu,Fx_n))} \varphi(t)\,dt \leqslant 0.$$

Since F is continuous, we get at infinity

$$\int_0^{F(\delta(gu,Gu),0,0,\delta(gu,Gu),\delta(gu,Gu),0)} \varphi(t)\,dt \leqslant 0$$

which implies, by using condition (F_a), $\delta(gu, Gu) = 0$; i.e., $Gu = \{gu\} = \{t\}$. Since the pair (g, G) is subcompatible, it follows that $Ggu = gGu$; i.e., $Gt = \{gt\}$. If $t \neq gt$, using (2) we have

$$\int_0^{F(\delta(Fx_n,Gt),d(fx_n,gt),\delta(fx_n,Fx_n),\delta(gt,Gt),\delta(fx_n,Gt),\delta(gt,Fx_n))} \varphi(t)\,dt \leqslant 0.$$

Taking limit as $n \to \infty$, we get

$$\int_0^{F(d(t,gt),d(t,gt),0,0,d(t,gt),d(gt,t))} \varphi(t)\,dt \leqslant 0,$$

which contradicts (F_1). Hence, $Gt = \{gt\} = \{t\}$. Since $G\mathcal{X} \subseteq f\mathcal{X}$, there is $v \in \mathcal{X}$ such that $\{t\} = Gt = \{fv\}$. If $Fv \neq \{t\}$, using (2) again, we have

$$\begin{aligned} &\int_0^{F(\delta(Fv,Gt),d(fv,gt),\delta(fv,Fv),\delta(gt,Gt),\delta(fv,Gt),\delta(gt,Fv))} \varphi(t)\,dt \\ = &\int_0^{F(\delta(Fv,t),0,\delta(t,Fv),0,0,\delta(t,Fv))} \varphi(t)\,dt \leqslant 0, \end{aligned}$$

which implies by using condition (F_b) that $\delta(Fv, t) = 0$, hence, $Fv = \{t\} = \{fv\}$. Since F and f are subcompatible, it follows that $Ffv = fFv$; i.e., $Ft = \{ft\}$. If $t \neq ft$, using (2) we have

$$\begin{aligned} &\int_0^{F(\delta(Ft,Gt),d(ft,gt),\delta(ft,Ft),\delta(gt,Gt),\delta(ft,Gt),\delta(gt,Ft))} \varphi(t)\,dt \\ = &\int_0^{F(d(ft,t),d(ft,t),0,0,d(ft,t),d(t,ft))} \varphi(t)\,dt \leqslant 0, \end{aligned}$$

which contradicts (F_1). Thus, $\{ft\} = \{t\} = Ft$.

We get the same conclusion if we use $(3')$ instead of (3).

The uniqueness of the common fixed point follows easily from conditions (2) and (F_1). □

Corollary 2.2. *Let f be a map from a metric space $(\mathcal{X}, d)$ into itself and let F be a map from $\mathcal{X}$ into $B(\mathcal{X})$. If*

(i) $F\mathcal{X} \subseteq f\mathcal{X}$,

(ii) f and F are subcompatible D-maps,

(iii)

$$\int_0^{F(\delta(Fx,Fy),d(fx,fy),\delta(fx,Fx),\delta(fy,Fy),\delta(fx,Fy),\delta(fy,Fx))} \varphi(t)\,dt \leqslant 0$$

for all x, y in $\mathcal{X}$, where $\varphi \in \Phi$ and F is continuous and satisfies conditions (F_a) and (F_1) or (F_b) and (F_1). If $F\mathcal{X}$ is closed, then, f and F have a unique common fixed point in $\mathcal{X}$.

The next Theorem is a generalization of Theorem 2.1.

Theorem 2.3. *Let f, g be self-maps of a metric space $(\mathcal{X}, d)$ and let $F_n\colon \mathcal{X} \to B(\mathcal{X})$, where $n = 1, 2, \ldots$ be multivalued maps such that*

(i) $F_n\mathcal{X} \subseteq g\mathcal{X}$ and $F_{n+1}\mathcal{X} \subseteq f\mathcal{X}$,

(ii)

$$\int_0^{F(\delta(F_nx,F_{n+1}y),d(fx,gy),\delta(fx,F_nx),\delta(gy,F_{n+1}y),\delta(fx,F_{n+1}y),\delta(gy,F_nx))} \varphi(t)\,dt \leqslant 0$$

for all x, y in $\mathcal{X}$, where $F \in \mathcal{F}$ and $\varphi \in \Phi$. If either

(iii) f and F_n are subcompatible D-maps; g and F_{n+1} are subcompatible and $F_n\mathcal{X}$ is closed, or

(iii)$'$ g and F_{n+1} are subcompatible D-maps; f and F_n are subcompatible and $F_{n+1}\mathcal{X}$ is closed.

Then, f, g and F_n have a unique common fixed point $t \in \mathcal{X}$ such that

$$F_nt = \{ft\} = \{gt\} = \{t\}.$$

Now, let Ψ be the set of all maps $\psi\colon \mathbb{R}_+ \to \mathbb{R}_+$ such that ψ is a Lebesgue-integrable which is summable, nonnegative and satisfies $\int_0^\epsilon \psi(t)dt > 0$ for each $\epsilon > 0$.

In [3], a common fixed point theorem for a pair of generalized contraction self-maps and a pair of multivalued maps in a complete metric space was obtained. Our second main subject is to complement and improve the result of [3] by relaxing the notion of δ-compatibility to subcompatibility, removing the assumption of continuity imposed on at least one of the four maps and deleting some conditions required on the functions Φ, a, b and c by using an integral type in a metric space instead of a complete metric space.

Theorem 2.4. *Let f, g be self-maps of a metric space $(\mathcal{X}, d)$ and let F, G be maps from $\mathcal{X}$ into $B(\mathcal{X})$ satisfying the following conditions*

(1$'$) f and g are surjective,

(2$'$)

$$\begin{aligned}\int_0^{F(\delta(Fx,Gy))} \psi(t)dt \leqslant{}& a(d(fx,gy)) \int_0^{F(d(fx,gy))} \psi(t)dt \\ &+ b(d(fx,gy)) \int_0^{F(\delta(fx,Fx))+F(\delta(gy,Gy))} \psi(t)dt \\ &+ c(d(fx,gy)) \int_0^{\min\{F(\delta(fx,Gy)),F(\delta(gy,Fx))\}} \psi(t)dt\end{aligned}$$

for all x, y in $\mathcal{X}$, where $F\colon [0,\infty) \to [0,\infty)$ is an upper semi-continuous map such that $F(t) = 0$ iff $t = 0$; $a, b, c\colon [0,\infty) \to [0,1)$ are upper semi-continuous such that $a(t) + c(t) < 1$ for every $t > 0$ and $\psi \in \Psi$. If either

(3$'$) f and F are subcompatible D-maps; g and G are subcompatible, or

($3''$) g and G are subcompatible D-maps; f and F are subcompatible. Then, f, g, F and G have a unique common fixed point $t \in \mathcal{X}$ such that

$$Ft = Gt = \{ft\} = \{gt\} = \{t\}.$$

Proof. Suppose that f and F are D-maps, then, there is a sequence $\{x_n\}$ in $\mathcal{X}$ such that $\lim_{n\to\infty} fx_n = t$ and $\lim_{n\to\infty} Fx_n = \{t\}$ for some $t \in \mathcal{X}$. By condition ($1'$), there exist points u, v in $\mathcal{X}$ such that $t = fu = gv$. First, we show that $Gv = \{gv\} = \{t\}$. Using inequality ($2'$) we get

$$\begin{aligned}
&\int_0^{F(\delta(Fx_n,Gv))} \psi(t)dt \\
&\leqslant a\left(d\left(fx_n, gv\right)\right) \int_0^{F(d(fx_n,gv))} \psi(t)dt \\
&+ b\left(d\left(fx_n, gv\right)\right) \int_0^{F(\delta(fx_n,Fx_n))+F(\delta(gv,Gv))} \psi(t)dt \\
&+ c\left(d\left(fx_n, gv\right)\right) \int_0^{\min\{F(\delta(fx_n,Gv)),F(\delta(gv,Fx_n))\}} \psi(t)dt.
\end{aligned}$$

Taking the limit as $n \to \infty$, one obtains

$$\int_0^{F(\delta(gv,Gv))} \psi(t)dt \leqslant b\left(0\right) \int_0^{F(\delta(gv,Gv))} \psi(t)dt < \int_0^{F(\delta(gv,Gv))} \psi(t)dt$$

this contradiction implies that $Gv = \{gv\} = \{t\}$. Since the pair (g, G) is subcompatible, then, $Ggv = gGv$; i.e., $Gt = \{gt\}$. We claim that $Gt = \{gt\} = \{t\}$. Suppose not, then, by condition ($2'$) we have

$$\begin{aligned}
\int_0^{F(\delta(Fx_n,Gt))} \psi(t)dt &\leqslant a\left(d\left(fx_n, gt\right)\right) \int_0^{F(d(fx_n,gt))} \psi(t)dt \\
&+ b\left(d\left(fx_n, gt\right)\right) \int_0^{F(\delta(fx_n,Fx_n))+F(\delta(gt,Gt))} \psi(t)dt \\
&+ c\left(d\left(fx_n, gt\right)\right) \int_0^{\min\{F(\delta(fx_n,Gt)),F(\delta(gt,Fx_n))\}} \psi(t)dt.
\end{aligned}$$

When $n \to \infty$ we obtain

$$\begin{aligned}
\int_0^{F(\delta(t,Gt))} \psi(t)dt &= \int_0^{F(d(t,gt))} \psi(t)dt \\
&\leqslant \left[a\left(d\left(t, gt\right)\right) + c\left(d\left(t, gt\right)\right)\right] \int_0^{F(d(t,gt))} \psi(t)dt \\
&< \int_0^{F(d(t,gt))} \psi(t)dt
\end{aligned}$$

which is a contradiction. Hence, $\{gt\} = \{t\} = Gt$. Next, we claim that $Fu = \{fu\} = \{t\}$. If not, then, by $(2')$ we get

$$\begin{aligned}\int_0^{F(\delta(Fu,fu))} \psi(t)dt &= \int_0^{F(\delta(Fu,Gt))} \psi(t)dt \\ &\leqslant a\left(d\left(fu,gt\right)\right) \int_0^{F(d(fu,gt))} \psi(t)dt \\ &+ b\left(d\left(fu,gt\right)\right) \int_0^{F(\delta(fu,Fu))+F(\delta(gt,Gt))} \psi(t)dt \\ &+ c\left(d\left(fu,gt\right)\right) \int_0^{\min\{F(\delta(fu,Gt)),F(\delta(gt,Fu))\}} \psi(t)dt \\ &= b\left(0\right) \int_0^{F(\delta(fu,Fu))} \psi(t)dt < \int_0^{F(\delta(fu,Fu))} \psi(t)dt\end{aligned}$$

which is a contradiction. Thus, $Fu = \{fu\} = \{t\}$. Since F and f are subcompatible, then, $Ffu = fFu$; i.e., $Ft = \{ft\}$. Suppose that $ft \neq t$. Then, the use of $(2')$ gives

$$\begin{aligned}\int_0^{F(d(ft,t))} \psi(t)dt &= \int_0^{F(\delta(Ft,Gt))} \psi(t)dt \\ &\leqslant a\left(d\left(ft,gt\right)\right) \int_0^{F(d(ft,gt))} \psi(t)dt \\ &+ b\left(d\left(ft,gt\right)\right) \int_0^{F(\delta(ft,Ft))+F(\delta(gt,Gt))} \psi(t)dt \\ &+ c\left(d\left(ft,gt\right)\right) \int_0^{\min\{F(\delta(ft,Gt)),F(\delta(gt,Ft))\}} \psi(t)dt \\ &= \left[a\left(d\left(ft,t\right)\right) + c\left(d\left(ft,t\right)\right)\right] \int_0^{F(d(ft,t))} \psi(t)dt \\ &< \int_0^{F(d(ft,t))} \psi(t)dt\end{aligned}$$

this contradiction implies that $ft = t$ and hence $Ft = \{ft\} = \{t\}$. Therefore t is a common fixed point of both f, g, F and G.

The uniqueness of the common fixed point follows easily from condition $(2')$.

We get the same conclusion if we consider $(3'')$ in lieu of $(3')$. □

Remark 2.5. Theorem 3.1 of [3] becomes a special case of Theorem 2.4 with $\psi(x) = 1$.

If we put $f = g$ in Theorem 2.4, we get the next corollary.

Corollary 2.6. *Let $(\mathcal{X}, d)$ be a metric space and let $f\colon \mathcal{X} \to \mathcal{X}$; $F, G\colon \mathcal{X} \to B(\mathcal{X})$ be maps. Suppose that*

(*i*) f *is surjective,*
(*ii*)

$$\begin{aligned}\int_0^{F(\delta(Fx,Gy))}\psi(t)dt &\leqslant a\left(d\left(fx,fy\right)\right)\int_0^{F(d(fx,fy))}\psi(t)dt\\ &+ b\left(d\left(fx,fy\right)\right)\int_0^{F(\delta(fx,Fx))+F(\delta(fy,Gy))}\psi(t)dt\\ &+ c\left(d\left(fx,fy\right)\right)\int_0^{\min\{F(\delta(fx,Gy)),F(\delta(fy,Fx))\}}\psi(t)dt\end{aligned}$$

for all x, y *in* $\mathcal{X}$*, where* F, ψ, a, b, c *are as in Theorem 2.4. If either*
(*iii*) f *and* F *are subcompatible D-maps;* f *and* G *are subcompatible, or*
(*iii*)$'$ f *and* G *are subcompatible D-maps;* f *and* F *are subcompatible.*
Then, f, F *and* G *have a unique common fixed point* $t \in \mathcal{X}$ *such that*

$$Ft = Gt = \{ft\} = \{t\}.$$

For a single map $f\colon \mathcal{X} \to \mathcal{X}$ (resp. a multivalued map $F\colon \mathcal{X} \to B(\mathcal{X})$), $\mathcal{F}_f$ (resp. $\mathcal{F}_F$) will denote the set of fixed point of f (resp. F).

Theorem 2.7. *Let* $F, G\colon \mathcal{X} \to B(\mathcal{X})$ *be multivalued maps and let* $f, g\colon \mathcal{X} \to \mathcal{X}$ *be single maps on the metric space* $\mathcal{X}$*. If inequality* $(2')$ *holds for all* x, y *in* $\mathcal{X}$*, then,*

$$(\mathcal{F}_f \cap \mathcal{F}_g) \cap \mathcal{F}_F = (\mathcal{F}_f \cap \mathcal{F}_g) \cap \mathcal{F}_G.$$

Proof. We can check the above equality by using inequality $(2')$. □

Theorems 2.4 and 2.7 imply the next one.

Theorem 2.8. *Let* f, g *be self-maps of a metric space* $(\mathcal{X}, d)$ *and let* F_n*, where* $n = 1, 2, \ldots$ *be maps from* $\mathcal{X}$ *into* $B(\mathcal{X})$ *such that*
(*i*) f *and* g *are surjective,*
(*ii*)

$$\begin{aligned}&\int_0^{F(\delta(F_nx,F_{n+1}y))}\psi(t)dt\\ &\leqslant a\left(d\left(fx,gy\right)\right)\int_0^{F(d(fx,gy))}\psi(t)dt\\ &+ b\left(d\left(fx,gy\right)\right)\int_0^{F(\delta(fx,F_nx))+F(\delta(gy,F_{n+1}y))}\psi(t)dt\\ &+ c\left(d\left(fx,gy\right)\right)\int_0^{\min\{F(\delta(fx,F_{n+1}y)),F(\delta(gy,F_nx))\}}\psi(t)dt\end{aligned}$$

for all x, y *in* $\mathcal{X}$*, where* F, ψ, a, b, c *are as in Theorem 2.4. If either*
(*iii*) f *and* F_1 *are subcompatible D-maps;* g *and* F_2 *are subcompatible, or*

(iii)′ g and F_2 are subcompatible D-maps; f and F_1 are subcompatible.

Then, f, g and F_n have a unique common fixed point $t \in \mathcal{X}$ such that

$$F_n t = \{ft\} = \{gt\} = \{t\} \quad \text{for} \quad n = 1, 2, \dots \ .$$

Let Ω be the family of all maps $\omega\colon \mathbb{R}_+ \to \mathbb{R}_+$ such that ω is upper semi-continuous and $\omega(t) < t$ for each $t > 0$.

In [1], Djoudi and Aliouche proved a common fixed point theorem of Greguš type for four maps satisfying a contractive condition of integral type in a metric space using the concept of weak compatibility. Our aim henceforth is to extend this result to multivalued maps by using the concept of D-maps.

Theorem 2.9. *Let $(\mathcal{X}, d)$ be a metric space and let $f, g\colon \mathcal{X} \to \mathcal{X}$; $F_k\colon \mathcal{X} \to B(\mathcal{X})$ be single and multivalued maps, respectively. Suppose that*

(i) $F_k\mathcal{X} \subseteq g\mathcal{X}$ and $F_{k+1}\mathcal{X} \subseteq f\mathcal{X}$,

(ii)

$$\begin{aligned}
&\left(\int_0^{\delta(F_k x, F_{k+1} y)} \psi(t)dt\right)^p \\
&\leqslant \omega\left(a\left(\int_0^{d(fx,gy)} \psi(t)dt\right)^p + (1-a)\max\left\{\alpha\left(\int_0^{\delta(fx,F_k x)} \psi(t)dt\right)^p,\right.\right. \\
&\beta\left(\int_0^{\delta(gy,F_{k+1} y)} \psi(t)dt\right)^p, \left(\int_0^{\delta(fx,F_k x)} \psi(t)dt\right)^{\frac{p}{2}} \left(\int_0^{\delta(gy,F_k x)} \psi(t)dt\right)^{\frac{p}{2}}, \\
&\left(\int_0^{\delta(gy,F_k x)} \psi(t)dt\right)^{\frac{p}{2}} \left(\int_0^{\delta(fx,F_{k+1} y)} \psi(t)dt\right)^{\frac{p}{2}}, \\
&\left.\left.\frac{1}{2}\left(\left(\int_0^{\delta(fx,F_k x)} \psi(t)dt\right)^p + \left(\int_0^{\delta(gy,F_{k+1} y)} \psi(t)dt\right)^p\right)\right\}\right)
\end{aligned}$$

for all x, y in $\mathcal{X}$, where $k \in \mathbb{N}^* = \{1, 2, \dots\}$, $\omega \in \Omega$, $\psi \in \Psi$, $0 < a < 1$, $0 < \alpha, \beta \leqslant 1$ and p is an integer such that $p \geqslant 1$. If either

(iii) f and F_k are subcompatible D-maps; g and F_{k+1} are subcompatible and $F_k\mathcal{X}$ is closed, or

(iii)′ g and F_{k+1} are subcompatible D-maps; f and F_k are subcompatible and $F_{k+1}\mathcal{X}$ is closed.

Then, f, g and F_k have a unique common fixed point $t \in \mathcal{X}$ such that

$$F_k t = \{ft\} = \{gt\} = \{t\}\,.$$

Proof. Suppose that f and F_k are D-maps, then, there exists a sequence $\{x_n\}$ in $\mathcal{X}$ such that $\lim_{n\to\infty} fx_n = t$ and $\lim_{n\to\infty} F_k x_n = \{t\}$ for some $t \in \mathcal{X}$. Since $F_k\mathcal{X}$ is closed and $F_k\mathcal{X} \subseteq g\mathcal{X}$, then, there is $u \in \mathcal{X}$ such that $gu = t$. If $F_{k+1}u \neq \{gu\}$,

using inequality (ii) we get

$$\left(\int_0^{\delta(F_kx_n,F_{k+1}u)}\psi(t)dt\right)^p$$
$$\leqslant \omega\left(a\left(\int_0^{d(fx_n,gu)}\psi(t)dt\right)^p\right.$$
$$+(1-a)\max\left\{\alpha\left(\int_0^{\delta(fx_n,F_kx_n)}\psi(t)dt\right)^p,\beta\left(\int_0^{\delta(gu,F_{k+1}u)}\psi(t)dt\right)^p,\right.$$
$$\left(\int_0^{\delta(fx_n,F_kx_n)}\psi(t)dt\right)^{\frac{p}{2}}\left(\int_0^{\delta(gu,F_kx_n)}\psi(t)dt\right)^{\frac{p}{2}},$$
$$\left(\int_0^{\delta(gu,F_kx_n)}\psi(t)dt\right)^{\frac{p}{2}}\left(\int_0^{\delta(fx_n,F_{k+1}u)}\psi(t)dt\right)^{\frac{p}{2}},$$
$$\left.\left.\left.\frac{1}{2}\left(\left(\int_0^{\delta(fx_n,F_kx_n)}\psi(t)dt\right)^p+\left(\int_0^{\delta(gu,F_{k+1}u)}\psi(t)dt\right)^p\right)\right\}\right).$$

Letting $n\to\infty$ we obtain

$$\left(\int_0^{\delta(gu,F_{k+1}u)}\psi(t)dt\right)^p$$
$$\leqslant \omega\left((1-a)\max\left\{\beta,\frac{1}{2}\right\}\left(\int_0^{\delta(gu,F_{k+1}u)}\psi(t)dt\right)^p\right)$$
$$<(1-a)\max\left\{\beta,\frac{1}{2}\right\}\left(\int_0^{\delta(gu,F_{k+1}u)}\psi(t)dt\right)^p<\left(\int_0^{\delta(gu,F_{k+1}u)}\psi(t)dt\right)^p$$

which is a contradiction. Then $F_{k+1}u=\{gu\}=\{t\}$. Since the pair (g,F_{k+1}) is subcompatible, we have $F_{k+1}gu=gF_{k+1}u$; i.e., $F_{k+1}t=\{gt\}$. If $t\neq gt$, using inequality (ii) we obtain

$$\left(\int_0^{\delta(F_kx_n,F_{k+1}t)}\psi(t)dt\right)^p$$
$$\leqslant \omega\left(a\left(\int_0^{d(fx_n,gt)}\psi(t)dt\right)^p\right.$$
$$+(1-a)\max\left\{\alpha\left(\int_0^{\delta(fx_n,F_kx_n)}\psi(t)dt\right)^p,\beta\left(\int_0^{\delta(gt,F_{k+1}t)}\psi(t)dt\right)^p,\right.$$
$$\left(\int_0^{\delta(fx_n,F_kx_n)}\psi(t)dt\right)^{\frac{p}{2}}\left(\int_0^{\delta(gt,F_kx_n)}\psi(t)dt\right)^{\frac{p}{2}},$$

$$\left(\int_0^{\delta(gt,F_kx_n)}\psi(t)dt\right)^{\frac{p}{2}}\left(\int_0^{\delta(fx_n,F_{k+1}t)}\psi(t)dt\right)^{\frac{p}{2}},$$
$$\frac{1}{2}\left(\left(\int_0^{\delta(fx_n,F_kx_n)}\psi(t)dt\right)^{p}+\left(\int_0^{\delta(gt,F_{k+1}t)}\psi(t)dt\right)^{p}\right)\Bigg\}\Bigg).$$

At infinity we get

$$\left(\int_0^{d(t,gt)}\psi(t)dt\right)^{p}\leqslant\omega\left(\left(\int_0^{d(t,gt)}\psi(t)dt\right)^{p}\right)<\left(\int_0^{d(t,gt)}\psi(t)dt\right)^{p}$$

which is a contradiction. Therefore $F_{k+1}t=\{gt\}=\{t\}$. Since $F_{k+1}\mathcal{X}\subseteq f\mathcal{X}$, there exists $v\in\mathcal{X}$ such that $F_{k+1}t=\{t\}=\{fv\}$. We claim that $F_kv=\{fv\}$, suppose not, then by condition (ii) we have

$$\left(\int_0^{\delta(F_kv,F_{k+1}t)}\psi(t)dt\right)^{p}$$
$$\leqslant\omega\left(a\left(\int_0^{d(fv,gt)}\psi(t)dt\right)^{p}+(1-a)\max\left\{\alpha\left(\int_0^{\delta(fv,F_kv)}\psi(t)dt\right)^{p},\right.\right.$$
$$\beta\left(\int_0^{\delta(gt,F_{k+1}t)}\psi(t)dt\right)^{p},\left(\int_0^{\delta(fv,F_kv)}\psi(t)dt\right)^{\frac{p}{2}}\left(\int_0^{\delta(gt,F_kv)}\psi(t)dt\right)^{\frac{p}{2}},$$
$$\left(\int_0^{\delta(gt,F_kv)}\psi(t)dt\right)^{\frac{p}{2}}\left(\int_0^{\delta(fv,F_{k+1}t)}\psi(t)dt\right)^{\frac{p}{2}},$$
$$\left.\left.\frac{1}{2}\left(\left(\int_0^{\delta(fv,F_kv)}\psi(t)dt\right)^{p}+\left(\int_0^{\delta(gt,F_{k+1}t)}\psi(t)dt\right)^{p}\right)\right\}\right),$$

that is,

$$\begin{aligned}\left(\int_0^{\delta(F_kv,fv)}\psi(t)dt\right)^{p}&\leqslant\omega\left((1-a)\left(\int_0^{\delta(F_kv,fv)}\psi(t)dt\right)^{p}\right)\\&<(1-a)\left(\int_0^{\delta(F_kv,fv)}\psi(t)dt\right)^{p}\\&<\left(\int_0^{\delta(F_kv,fv)}\psi(t)dt\right)^{p}\end{aligned}$$

which is a contradiction. Hence $F_kv=\{fv\}=\{t\}$. Since the pair (f,F_k) is subcompatible, then, $F_kfv=fF_kv$; i.e., $F_kt=\{ft\}$. The use of (ii) gives

$$\left(\int_0^{\delta(F_kt,F_{k+1}t)}\psi(t)dt\right)^{p}$$

$$\leqslant \omega\left(a\left(\int_0^{d(ft,gt)}\psi(t)dt\right)^p + (1-a)\max\left\{\alpha\left(\int_0^{\delta(ft,F_kt)}\psi(t)dt\right)^p,\right.\right.$$
$$\beta\left(\int_0^{\delta(gt,F_{k+1}t)}\psi(t)dt\right)^p, \left(\int_0^{\delta(ft,F_kt)}\psi(t)dt\right)^{\frac{p}{2}}\left(\int_0^{\delta(gt,F_kt)}\psi(t)dt\right)^{\frac{p}{2}},$$
$$\left(\int_0^{\delta(gt,F_kt)}\psi(t)dt\right)^{\frac{p}{2}}\left(\int_0^{\delta(ft,F_{k+1}t)}\psi(t)dt\right)^{\frac{p}{2}},$$
$$\left.\left.\frac{1}{2}\left(\left(\int_0^{\delta(ft,F_kt)}\psi(t)dt\right)^p + \left(\int_0^{\delta(gt,F_{k+1}t)}\psi(t)dt\right)^p\right)\right\}\right),$$

i.e.,

$$\left(\int_0^{d(ft,t)}\psi(t)dt\right)^p \leqslant \omega\left(\left(\int_0^{d(ft,t)}\psi(t)dt\right)^p\right) < \left(\int_0^{d(ft,t)}\psi(t)dt\right)^p$$

this contradiction implies that $\{ft\} = \{t\} = F_kt$. Thus, t is a common fixed point of f, g and F_k.

The uniqueness of the common fixed point follows from inequality (ii).

If one uses condition $(iii)'$ instead of (iii), one gets the same conclusion. □

Theorem 2.10. *Let $(\mathcal{X}, d)$ be a metric space and let $f, g\colon \mathcal{X} \to \mathcal{X}$; $F_n\colon \mathcal{X} \to B(\mathcal{X})$ be single and multivalued maps such that*

(i) $F_n\mathcal{X} \subseteq g\mathcal{X}$ and $F_{n+1}\mathcal{X} \subseteq f\mathcal{X}$,

(ii)

$$\left(\int_0^{\delta(F_nx,F_{n+1}y)}\psi(t)dt\right)^p$$
$$\leqslant \omega\left(a\left(\int_0^{d(fx,gy)}\psi(t)dt\right)^p + (1-a)\max\left\{\int_0^{\delta(fx,F_nx)}\psi(t)dt,\right.\right.$$
$$\int_0^{\delta(gy,F_{n+1}y)}\psi(t)dt, \left(\int_0^{\delta(fx,F_nx)}\psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(gy,F_nx)}\psi(t)dt\right)^{\frac{1}{2}},$$
$$\left.\left.\left(\int_0^{\delta(gy,F_nx)}\psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(fx,F_{n+1}y)}\psi(t)dt\right)^{\frac{1}{2}}\right\}^p\right)$$

for all x, y in $\mathcal{X}$, where $\omega \in \Omega$, $\psi \in \Psi$, $0 < a < 1$ and p is an integer such that $p \geqslant 1$. If either

(iii) f and F_n are subcompatible D-maps; g and F_{n+1} are subcompatible and $F_n\mathcal{X}$ is closed, or

$(iii)'$ g and F_{n+1} are subcompatible D-maps; f and F_n are subcompatible and $F_{n+1}\mathcal{X}$ is closed.

Then, f, g and F_n have a unique common fixed point $t \in \mathcal{X}$ such that

$$F_n t = \{ft\} = \{gt\} = \{t\} \quad \text{for} \quad n = 1, 2, \dots \ .$$

Proof. It is similar to the proof of Theorem 2.9. □

Now, we prove a unique common fixed point theorem of Greguš type by using a strict contractive condition of integral type for two pairs of single and multivalued maps in a metric space.

Theorem 2.11. *Let f and g be self-maps of a metric space $(\mathcal{X}, d)$ and let $\{F_n\}$, $n = 1, 2, \dots$ be multivalued maps from $\mathcal{X}$ into $B(\mathcal{X})$ such that*

$(1'')$ f and g are surjective,

$(2'')$

$$\begin{aligned}
&\int_0^{\delta(F_1x, F_ky)} \psi(t)dt \\
&< \alpha \int_0^{d(fx,gy)} \psi(t)dt + (1-\alpha) \max \left\{ a \int_0^{\delta(fx, F_1x)} \psi(t)dt, \right. \\
&b \int_0^{\delta(gy, F_ky)} \psi(t)dt, c \left(\int_0^{\delta(fx, F_1x)} \psi(t)dt \right)^{\frac{1}{2}} \left(\int_0^{\delta(gy, F_1x)} \psi(t)dt \right)^{\frac{1}{2}}, \\
&\left. d \left(\int_0^{\delta(gy, F_1x)} \psi(t)dt \right)^{\frac{1}{2}} \left(\int_0^{\delta(fx, F_ky)} \psi(t)dt \right)^{\frac{1}{2}} \right\}
\end{aligned}$$

for all x, y in $\mathcal{X}$ and some $k > 1$ for which the right hand side is positive, where $\psi \in \Psi$, $0 < \alpha, a, b, c, d < 1$ and $\alpha + d(1-\alpha) < 1$. If either

$(3'')$ f and F_1 are subcompatible D-maps; g and F_k are subcompatible, or

$(3''')$ g and F_k are subcompatible D-maps; f and F_1 are subcompatible.

Then, f, g and $\{F_n\}$ have a unique common fixed point $t \in \mathcal{X}$ such that

$$F_n t = \{ft\} = \{gt\} = \{t\}, \quad \text{for} \quad n = 1, 2, \dots \ .$$

Proof. Suppose that condition $(3'')$ holds, then, there is a sequence $\{x_n\}$ in $\mathcal{X}$ such that $fx_n \to t$ and $F_1x_n \to \{t\}$ as $n \to \infty$ for some $t \in \mathcal{X}$. By condition $(1'')$, there are two elements u and v in $\mathcal{X}$ such that $t = fu = gv$. We show that $\{t\} = F_kv$. Indeed, using inequality $(2'')$ we get

$$\begin{aligned}
&\int_0^{\delta(F_1x_n, F_kv)} \psi(t)dt \\
&< \alpha \int_0^{d(fx_n,gv)} \psi(t)dt + (1-\alpha) \max \left\{ a \int_0^{\delta(fx_n, F_1x_n)} \psi(t)dt, \right. \\
&b \int_0^{\delta(gv, F_kv)} \psi(t)dt, c \left(\int_0^{\delta(fx_n, F_1x_n)} \psi(t)dt \right)^{\frac{1}{2}} \left(\int_0^{\delta(gv, F_1x_n)} \psi(t)dt \right)^{\frac{1}{2}},
\end{aligned}$$

$$d\left(\int_0^{\delta(gv,F_1x_n)}\psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(fx_n,F_kv)}\psi(t)dt\right)^{\frac{1}{2}}\Bigg\}.$$

Taking limit as $n\to\infty$, we obtain

$$\int_0^{\delta(t,F_kv)}\psi(t)dt\leqslant b\,(1-\alpha)\int_0^{\delta(t,F_kv)}\psi(t)dt<\int_0^{\delta(t,F_kv)}\psi(t)dt$$

thus, we have $F_kv=\{t\}=\{gv\}$ and since g and F_k are subcompatible, we have $F_kgv=gF_kv$; that is, $F_kt=\{gt\}$. Again, by $(2'')$ we obtain

$$\begin{aligned}&\int_0^{\delta(F_1x_n,F_kt)}\psi(t)dt\\&<\alpha\int_0^{d(fx_n,gt)}\psi(t)dt+(1-\alpha)\max\Bigg\{a\int_0^{\delta(fx_n,F_1x_n)}\psi(t)dt,\\&b\int_0^{\delta(gt,F_kt)}\psi(t)dt,c\left(\int_0^{\delta(fx_n,F_1x_n)}\psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(gt,F_1x_n)}\psi(t)dt\right)^{\frac{1}{2}},\\&d\left(\int_0^{\delta(gt,F_1x_n)}\psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(fx_n,F_kt)}\psi(t)dt\right)^{\frac{1}{2}}\Bigg\}.\end{aligned}$$

When $n\to\infty$, we get

$$\int_0^{d(t,gt)}\psi(t)dt\leqslant[\alpha+d\,(1-\alpha)]\int_0^{d(t,gt)}\psi(t)dt<\int_0^{d(t,gt)}\psi(t)dt$$

this contradiction implies that $\{t\}=\{gt\}=F_kt=\{fu\}$. We claim that $F_1u=\{t\}$. By condition $(2'')$ we have

$$\begin{aligned}&\int_0^{\delta(F_1u,t)}\psi(t)dt=\int_0^{\delta(F_1u,F_kt)}\psi(t)dt\\&<\alpha\int_0^{d(fu,gt)}\psi(t)dt+(1-\alpha)\max\Bigg\{a\int_0^{\delta(fu,F_1u)}\psi(t)dt,\\&b\int_0^{\delta(gt,F_kt)}\psi(t)dt,c\left(\int_0^{\delta(fu,F_1u)}\psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(gt,F_1u)}\psi(t)dt\right)^{\frac{1}{2}},\\&d\left(\int_0^{\delta(gt,F_1u)}\psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(fu,F_kt)}\psi(t)dt\right)^{\frac{1}{2}}\Bigg\}\\&=(1-\alpha)\max\{a,c\}\int_0^{\delta(F_1u,t)}\psi(t)dt<\int_0^{\delta(F_1u,t)}\psi(t)dt\end{aligned}$$

this contradiction demands that $F_1u = \{t\} = \{fu\}$. Since f and F_1 are subcompatible, then, $F_1fu = fF_1u$; that is, $F_1t = \{ft\}$. Moreover, by $(2'')$ one may get

$$\begin{aligned}
&\int_0^{d(ft,t)} \psi(t)dt = \int_0^{\delta(F_1t,F_kt)} \psi(t)dt \\
&< \alpha \int_0^{d(ft,gt)} \psi(t)dt + (1-\alpha)\max\left\{ a\int_0^{\delta(ft,F_1t)} \psi(t)dt,\right. \\
&b\int_0^{\delta(gt,F_kt)} \psi(t)dt, c\left(\int_0^{\delta(ft,F_1t)} \psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(gt,F_1t)} \psi(t)dt\right)^{\frac{1}{2}}, \\
&\left. d\left(\int_0^{\delta(gt,F_1t)} \psi(t)dt\right)^{\frac{1}{2}}\left(\int_0^{\delta(ft,F_kt)} \psi(t)dt\right)^{\frac{1}{2}}\right\} \\
&= [\alpha + d\,(1-\alpha)]\int_0^{d(ft,t)} \psi(t)dt < \int_0^{d(ft,t)} \psi(t)dt
\end{aligned}$$

which is a contradiction. Thus, $\{ft\} = \{t\} = F_1t$. Therefore, $F_1t = F_kt = \{ft\} = \{gt\} = \{t\}$.

Uniqueness follows easily from condition $(2'')$. The proof is thus completed. □

Important remark. Every contractive or strict contractive condition of integral type automatically includes a corresponding contractive or strict contractive condition, not involving integrals, by setting $\varphi(t) = 1$ (resp. $\psi(t) = 1$) over $\mathbb{R}_+$. So, our results extend, generalize and complement several various results existing in the literature.

References

[1] Djoudi, A., Aliouche, A., Common fixed point theorems of Greguš type for weakly compatible mappings satisfying contractive conditions of integral type, *J. Math. Anal. Appl.* 329 (2007) no. 1, 31–45.

[2] Djoudi, A., Khemis, R., Fixed points for set and single valued maps without continuity, *Demonstratio Math.* 38 (2005) no. 3, 739–751.

[3] Elamrani, M., Mehdaoui, B., Common fixed point theorems for compatible and weakly compatible mappings, *Rev. Colombiana Mat.* 34 (2000) no. 1, 25–33.

[4] Fisher, B., Common fixed points of mappings and set-valued mappings, *Rostock. Math. Kolloq.* 18 (1981) 69–77.

[5] Fisher, B., Sessa, S., Two common fixed point theorems for weakly commuting mappings, *Period. Math. Hungar.* 20 (1989) no. 3, 207–218.

[6] Jungck, G., Compatible mappings and common fixed points, *Internat. J. Math. Math. Sci.* 9 (1986) no. 4, 771–779.

[7] Jungck, G., Common fixed points for noncontinuous nonself maps on nonmetric spaces, *Far East J. Math. Sci.* 4 (1996) no. 2, 199–215.

[8] Jungck, G., Murthy, P.P., Cho, Y.J., Compatible mappings of type (A) and common fixed points, *Math. Japon.* 38 (1993) no. 2, 381–390.

[9] Jungck, G., Rhoades, B.E., Some fixed point theorems for compatible maps, *Internat. J. Math. Math. Sci.* 16 (1993) no. 3, 417–428.

[10] Jungck, G., Rhoades, B.E., Fixed points for set valued functions without continuity, *Indian J. Pure Appl. Math.* 29 (1998) no. 3, 227–238.

[11] 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, *Matematiche (Catania)* 50 (1995) no. 1, 15–33.

[12] Pathak, H.K., Cho, Y.J., Kang, S.M., Madharia, B., Compatible mappings of type (C) and common fixed point theorems of Greguš type, *Demonstratio Math.* 31 (1998) no. 3, 499–518.

[13] Pathak, H.K., Khan, M.S., Compatible mappings of type (B) and common fixed point theorems of Greguš type, *Czechoslovak Math. J.* 45(120) (1995) no. 4, 685–698.

[14] Pathak, H.K., Tiwari, R., Khan, M.S., A common fixed point theorem satisfying integral type implicit relations, *Appl. Math. E-Notes* 7 (2007) 222–228.

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

H. Bouhadjera
A. Djoudi
Laboratoire de Mathématiques Appliquées
Université Badji Mokhtar
B. P. 12, 23000, Annaba
Algérie
e-mail: b_hakima2000@yahoo.fr

Annales Mathematicae et Informaticae
***35** (2008) pp. 61–73*
http://www.ektf.hu/ami

On Worley's theorem in Diophantine approximations*

Andrej Dujella[a], Bernadin Ibrahimpašić[b]

[a]Department of Mathematics, University of Zagreb

[b]Pedagogical Faculty, University of Bihać

Submitted 11 March 2008; Accepted 17 October 2008

Abstract

In this paper we prove several results on connection between continued fractions and rational approximations of the form $|\alpha - a/b| < k/b^2$, for a positive integer k.

Keywords: Continued fractions

MSC: Primary 11A55, 11J70; Secondary 11D09.

1. Introduction

The classical Legendre's theorem in Diophantine approximations states that if a real number α and a rational number $\frac{a}{b}$ (we will always assume that $b \geqslant 1$), satisfy the inequality

$$\left|\alpha - \frac{a}{b}\right| < \frac{1}{2b^2}, \tag{1.1}$$

then $\frac{a}{b}$ is a convergent of the continued fraction expansion of $\alpha = [a_0; a_1, \ldots]$. This result has been extended by Fatou [3] (see also [5, p.16]), who showed that if

$$|\alpha - \frac{a}{b}| < \frac{1}{b^2},$$

then $\frac{a}{b} = \frac{p_m}{q_m}$ or $\frac{p_{m+1} \pm p_m}{q_{m+1} \pm q_m}$, where $\frac{p_m}{q_m}$ denotes the m-th convergent of α.

In 1981, Worley [12] generalized these results to the inequality $\left|\alpha - \frac{a}{b}\right| < \frac{k}{b^2}$, where k is an arbitrary positive real number. Worley's result was slightly improved in [1].

*The first author was supported by the Ministry of Science, Education and Sports, Republic of Croatia, grant 037-0372781-2821.

Theorem 1.1 (Worley [12], Dujella [1]). *Let α be a real number and let a and b be coprime nonzero integers, satisfying the inequality*

$$\left|\alpha - \frac{a}{b}\right| < \frac{k}{b^2}, \tag{1.2}$$

where k is a positive real number. Then $(a,b) = (rp_{m+1} \pm sp_m, rq_{m+1} \pm sq_m)$, for some $m \geqslant -1$ and nonnegative integers r and s such that $rs < 2k$.

The original result of Worley [12, Theorem 1] contains three types of solutions to the inequality (1.2). Two types correspond to two possible choices for signs + and − in $(rp_{m+1} \pm sp_m, rq_{m+1} \pm sq_m)$, while [1, Theorem 1] shows that the third type (corresponding to the case $a_{m+2} = 1$) can be omitted.

In Section 3 we will show that Theorem 1.1 is sharp, in the sense that the condition $rs < 2k$ cannot be replaced by $rs < (2-\varepsilon)k$ for any $\varepsilon > 0$. However, it appears that the coefficients r and s show different behavior. So, improvements of Theorem 1.1 are possible if we allow nonsymmetric conditions on r and s. Indeed, already the paper of Worley [12] contains an important contribution in that direction.

Theorem 1.2 (Worley [12], Theorem 2). *If α is an irrational number, $k \geqslant \frac{1}{2}$ and $\frac{a}{b}$ is a rational approximation to α (in reduced form) for which the inequality (1.2) holds, then either $\frac{a}{b}$ is a convergent $\frac{p_m}{q_m}$ to α or $\frac{a}{b}$ has one of the following forms:*

$$\text{(i)}\ \ \frac{a}{b} = \frac{rp_{m+1}+sp_m}{rq_{m+1}+sq_m} \qquad \begin{array}{l} r > s \quad \text{and} \quad rs < 2k, \quad \text{or} \\ r \leqslant s \quad \text{and} \quad rs < k + \frac{r^2}{a_{m+2}}, \end{array}$$

$$\text{(ii)}\ \ \frac{a}{b} = \frac{sp_{m+1}-tp_m}{sq_{m+1}-tq_m} \qquad \begin{array}{l} s < t \quad \text{and} \quad st < 2k, \quad \text{or} \\ s \geqslant t \quad \text{and} \quad st\left(1 - \frac{t}{2s}\right) < k, \end{array}$$

where r, s and t are positive integers.

Since the fraction a/b is in reduced form, it is clear that in the statements of Theorems 1.1 and 1.2 we may assume that $\gcd(r,s) = 1$ and $\gcd(s,t) = 1$.

Worley [12, Corollary, p.206] also gave the explicit version of his result for $k = 2$: $|\alpha - \frac{a}{b}| < \frac{2}{b^2}$ implies $\frac{a}{b} = \frac{p_m}{q_m}$, $\frac{p_{m+1} \pm p_m}{q_{m+1} \pm q_m}$, $\frac{2p_{m+1} \pm p_m}{2q_{m+1} \pm q_m}$, $\frac{3p_{m+1}+p_m}{3q_{m+1}+q_m}$, $\frac{p_{m+1} \pm 2p_m}{q_{m+1} \pm 2q_m}$ or $\frac{p_{m+1}-3p_m}{q_{m+1}-3q_m}$. This result for $k = 2$ has been applied for solving some Diophantine equations. In [7], it was applied to the problem of finding positive integers a and b such that $(a^2+b^2)/(ab+1)$ is an integer, and in [2] it was used for solving the family of Thue inequalities

$$|x^4 - 4cx^3y + (6c+2)x^2y^2 + 4cxy^2 + y^4| \leqslant 6c + 4.$$

On the other hand, Theorem 1.1 has applications in cryptography, too. Namely, in [1], a modification of Verheul and van Tilborg variant of Wiener's attack ([10, 11]) on RSA cryptosystem with small secret exponent has been described, which is based on Theorem 1.1.

We will extend Worley's work and give explicit and sharp versions of Theorems 1.1 and 1.2 for $k = 3, 4, 5, \ldots, 12$. We will list the pairs (r, s) which appear in the expression of solutions of (1.2) in the form $(a, b) = (rp_{m+1} \pm sp_m, rq_{m+1} \pm sq_m)$, and we will show by explicit examples that all pairs from the list are indeed necessary. We hope that our results will also find applications on Diophantine problems, and in Section 4 we will present such an application. In such applications, it is especially of interest to have smallest possible list of pairs (r, s). It is certainly possible to extend our result for $k > 12$. However, already our results make it possible to reveal certain patterns, and they also suffice for our Diophantine applications.

2. Explicit versions of Worley's theorem

We start by few details from the proof of Theorem 1.1 in [1], which will be useful in our future arguments. In particular, we will explain how the integer m appearing in the statement of Theorem 1.1 can be found. We assume that $\alpha < \frac{a}{b}$, since the other case is completely analogous. Let m be the largest odd integer satisfying

$$\alpha < \frac{a}{b} \leqslant \frac{p_m}{q_m}.$$

If $\frac{a}{b} > \frac{p_1}{q_1}$, we take $m = -1$, following the convention that $p_{-1} = 1$, $q_{-1} = 0$. Since $|p_{m+1}q_m - p_m q_{m+1}| = 1$, the numbers r and s defined by

$$\begin{aligned} a &= rp_{m+1} + sp_m, \\ b &= rq_{m+1} + sq_m \end{aligned}$$

are integers, and since $\frac{p_{m+1}}{q_{m+1}} < \frac{a}{b} \leqslant \frac{p_m}{q_m}$, we have that $r \geqslant 0$ and $s > 0$. From the maximality of m, we find that

$$\frac{sa_{m+2} - r}{bq_{m+2}} = \left|\frac{p_{m+2}}{q_{m+2}} - \frac{a}{b}\right| < \left|\alpha - \frac{a}{b}\right| < \frac{k}{b^2}. \tag{2.1}$$

From (2.1) we immediately have

$$a_{m+2} > \frac{r}{s}, \tag{2.2}$$

and we can derive the inequality

$$r^2 - sra_{m+2} + ka_{m+2} > 0 \tag{2.3}$$

(see [1, proof of Theorem 1] for details, and note also that (2.3) is exactly the inequality from Theorem 1.2 (i) - the second case).

Let us define a positive integer t by $t = sa_{m+2} - r$. Then we have

$$\begin{aligned} a &= rp_{m+1} + sp_m = sp_{m+2} - tp_{m+1}, \\ b &= rq_{m+1} + sq_m = sq_{m+2} - tq_{m+1}, \end{aligned}$$

and s and t satisfy analogs of (2.2) and (2.3):

$$a_{m+2} > \frac{t}{s}, \tag{2.4}$$

$$t^2 - sta_{m+2} + ka_{m+2} > 0. \tag{2.5}$$

If $r > t$, i.e. $rs > st$, then we will represent a and b in terms of s and t (which corresponds to $-$ sign in Theorem 1.1).

Let us consider now the case $k = 3$. Hence, we are considering the inequality

$$|\alpha - \frac{a}{b}| < \frac{3}{b^2}. \tag{2.6}$$

By Theorem 1.1, we have that $(a, b) = (rp_{m+1} + sp_m, rq_{m+1} + sq_m)$ or $(sp_{m+2} - tp_{m+1}, sq_{m+2} - tq_{m+1})$, where $rs < 6$, $st < 6$, $\gcd(r, s) = 1$ and $\gcd(s, t) = 1$. However, the inequalities (2.3) and (2.5) for $r = 1$, resp. $t = 1$, show that the pairs $(r, s) = (1, 4), (1, 5)$ and $(s, t) = (4, 1), (5, 1)$ can be omitted. Therefore, we proved

Proposition 2.1. *If a real number α and a rational number $\frac{a}{b}$ satisfy the inequality (2.6), then $\frac{a}{b} = \frac{rp_{m+1} + sp_m}{rq_{m+1} + sq_m}$, where*

$$(r, s) \in R_3 = \{(0, 1), (1, 1), (1, 2), (1, 3), (2, 1), (3, 1), (4, 1), (5, 1)\},$$

or $\frac{a}{b} = \frac{sp_{m+2} - tp_{m+1}}{sq_{m+2} - tq_{m+1}}$, where

$$(s, t) \in T_3 = \{(1, 1), (2, 1), (3, 1), (1, 2), (1, 3), (1, 4), (1, 5)\}$$

(for an integer $m \geqslant -1$).

Our next aim is to show that Proposition 2.1 is sharp, i.e. that if we omit any of the pairs (r, s) or (s, t) appearing in Proposition 2.1, the statement of the proposition will no longer be valid. More precisely, if we omit a pair $(r', s') \in R_3$, then there exist a real number α and a rational number $\frac{a}{b}$ satisfying (2.6), but such that $\frac{a}{b}$ cannot be represented in the form $\frac{a}{b} = \frac{rp_{m+1}+sp_m}{rq_{m+1}+sq_m}$ nor $\frac{a}{b} = \frac{sp_{m+2}-tp_{m+1}}{sq_{m+2}-tq_{m+1}}$, where $m \geqslant -1$, $(r, s) \in R_3 \setminus \{(r', s')\}$, $(s, t) \in T_3$ (and similarly for an omitted pair $(s', t') \in T_3$).

We will show that by giving explicit examples for each pair. Although we have found many such examples of different form, in the next table we give numbers α of the form $\sqrt{d}$, where d is a non-square positive integer.

α	a	b	m	r	s	t
$\sqrt{10}$	3	1	0	**0**	**1**	6
$\sqrt{17}$	37	9	0	**1**	**1**	7
$\sqrt{2}$	5	4	0	**1**	**2**	3
$\sqrt{8}$	23	8	1	**1**	**3**	2
$\sqrt{17}$	70	17	0	**2**	**1**	6
$\sqrt{26}$	158	31	0	**3**	**1**	7
$\sqrt{26}$	209	41	0	**4**	**1**	6
$\sqrt{37}$	371	61	0	**5**	**1**	7

α	a	b	m	r	s	t
$\sqrt{17}$	235	57	0	7	**1**	**1**
$\sqrt{2}$	11	8	0	3	**2**	**1**
$\sqrt{8}$	37	13	1	2	**3**	**1**
$\sqrt{17}$	202	49	0	6	**1**	**2**
$\sqrt{26}$	362	71	0	7	**1**	**3**
$\sqrt{26}$	311	61	0	6	**1**	**4**
$\sqrt{37}$	517	85	0	7	**1**	**5**

For example, consider $\alpha = \sqrt{8} = [2, \overline{1,4}]$. Its rational approximation $\frac{23}{8}$ (the forth row of the table) satisfies $\left|\sqrt{8} - \frac{23}{8}\right| \approx 0.046572875 < \frac{3}{8^2}$. The convergents of $\sqrt{8}$ are $\frac{2}{1}$, $\frac{3}{1}$, $\frac{14}{5}$, $\frac{17}{6}$, $\frac{82}{29}$, $\frac{99}{35}$, $\frac{478}{169}$, ... The only representation of the fraction $\frac{23}{8}$ in the form $\frac{rp_{m+1}+sp_m}{rq_{m+1}+sq_m}$, $(r,s) \in R_3$ or $\frac{sp_{m+2}-tp_{m+1}}{sq_{m+2}-tq_{m+1}}$, $(s,t) \in T_3$ is $\frac{23}{8} = \frac{1\cdot 14+3\cdot 3}{1\cdot 5+3\cdot 1} = \frac{1\cdot p_2+3\cdot p_1}{1\cdot q_2+3\cdot q_1}$, which shows that the pair $(1,3)$ cannot be omitted from the set R_3.

Proposition 2.2. *Let $k \in \{4,5,6,7,8,9,10,11,12\}$. If a real number α and a rational number $\frac{a}{b}$ satisfy the inequality (1.2), then $\frac{a}{b} = \frac{rp_{m+1}+sp_m}{rq_{m+1}+sq_m}$, where $(r,s) \in R_k = R_{k-1} \cup R'_k$, or $\frac{a}{b} = \frac{sp_{m+2}-tp_{m+1}}{sq_{m+2}-tq_{m+1}}$, where $(s,t) \in T_k = T_{k-1} \cup T'_k$ (for an integer $m \geqslant -1$), where the sets R'_k and T'_k are given in the following table. Moreover, if any of the elements in sets R_k or T_k is omitted, the statement will no longer be valid.*

k	R'_k	T'_k
4	$\{(1,4),(3,2),(6,1),(7,1)\}$	$\{(4,1),(2,3),(1,6),(1,7)\}$
5	$\{(1,5),(2,3),(8,1),(9,1)\}$	$\{(5,1),(3,2),(1,8),(1,9)\}$
6	$\{(1,6),(5,2),(10,1),(11,1)\}$	$\{(6,1),(2,5),(1,10),(1,11)\}$
7	$\{(1,7),(2,5),(4,3),(12,1),(13,1)\}$	$\{(7,1),(5,2),(3,4),(1,12),(1,13)\}$
8	$\{(1,8),(3,4),(7,2),(14,1),(15,1)\}$	$\{(8,1),(4,3),(2,7),(1,14),(1,15)\}$
9	$\{(1,9),(5,3),(16,1),(17,1)\}$	$\{(9,1),(3,5),(1,16),(1,17)\}$
10	$\{(1,10),(9,2),(18,1),(19,1)\}$	$\{(10,1),(2,9),(1,18),(1,19)\}$
11	$\{(1,11),(2,7),(3,5),(20,1),(21,1)\}$	$\{(11,1),(7,2),(5,3),(1,20),(1,21)\}$
12	$\{(1,12),(5,4),(7,3),$ $(11,2),(22,1),(23,1)\}$	$\{(12,1),(4,5),(3,7),$ $(2,11),(1,22),(1,23)\}$

Proof. By Theorem 1.1, we have to consider only pairs of nonnegative integers (r,s) and (s,t) satisfying $rs < 2k$, $st < 2k$, $\gcd(r,s) = 1$ and $\gcd(s,t) = 1$. Furthermore, as in the case $k = 3$, it follows directly from the inequalities (2.3) and (2.5) for $r = 1$, resp. $t = 1$, that the pairs $(r,s) = (1,s)$ and $(s,t) = (s,1)$ with $s \geqslant k+1$ can be omitted. Similarly, for $r = 2$ or 3, resp. $t = 2$ or 3, we can exclude the pairs $(r,s) = (2,s)$ and $(s,t) = (s,2)$ with $s \geqslant \frac{k}{2} + 2$, and the pairs $(r,s) = (3,s)$ and $(s,t) = (s,3)$ with $s \geqslant \frac{k}{3} + 3$.

Now we show that all remaining possible pairs which are not listed in the statement of Proposition 2.2 can be replaced with other pairs with smaller products rs, resp. st. We give details only for pairs (r,s), since the proof for pairs (s,t) is completely analogous (using the inequalities (2.4) and (2.5), instead of (2.2) and (2.3)).

Consider the case $k = 4$ and $(r,s) = (2,3)$. By (2.3), we obtain $a_{m+2} < 2$. Thus, the pair $(r,s) = (2,3)$ can appear only for $a_{m+2} = 1$. However, in that case we have $t = sa_{m+2} - r = 1$, and therefore the $(r,s) = (2,3)$ can be replaced by the pair $(s,t) = (3,1)$.

Analogously we can show that for $k = 7$ the pair $(r,s) = (3,4)$ can be replaced by $(s,t) = (4,1)$, for $k = 8,9,10$ the pair $(r,s) = (3,5)$ can be replaced by $(s,t) = (5,2)$, while for $k = 11,12$ the pair $(r,s) = (4,5)$ can be replaced by $(s,t) = (5,1)$.

We have only three remaining pairs to consider: the pair $(r,s)=(5,3)$ for $k=8$ and the pairs $(r,s)=(5,4)$ and $(r,s)=(7,3)$ for $k=11$. For $(r,s)=(5,3)$ and $k=8$, from (2.2) and (2.3) we obtain $\frac{5}{3}<a_{m+2}<\frac{25}{7}$, and therefore we have two possibilities: $a_{m+2}=2$ or $a_{m+2}=3$. If $a_{m+2}=2$, we can replace $(r,s)=(5,3)$ by $(s,t)=(3,1)$, while if $a_{m+2}=3$, we can replace it by $(s,t)=(3,4)$. Similar approach works for two pairs with $k=11$. For $(r,s)=(5,4)$, from (2.2) and (2.3) we obtain $\frac{5}{4}<a_{m+2}<\frac{25}{9}$, which implies $a_{m+2}=2$. Then we have $t=3$ and the pair $(r,s)=(5,4)$ can be replaced by the pair $(s,t)=(4,3)$. For $(r,s)=(7,3)$ we obtain $\frac{7}{3}<a_{m+2}<\frac{49}{10}$, which yields $a_{m+2}=3$ or $a_{m+2}=4$. If $a_{m+2}=3$, we can replace $(r,s)=(7,3)$ by $(s,t)=(3,2)$, while if $a_{m+2}=4$, we can replace it by $(s,t)=(3,5)$.

It remains to show that all pairs listed in the statement of the proposition are indeed necessary (they cannot be omitted). This is shown by the examples from the following tables:

$k=4$						
α	a	b	m	r	s	t
$\sqrt{35}$	89	15	1	**1**	**4**	3
$\sqrt{39}$	968	155	1	**3**	**2**	5
$\sqrt{50}$	601	85	0	**6**	**1**	8
$\sqrt{65}$	911	113	0	**7**	**1**	9
$\sqrt{35}$	219	37	1	3	**4**	**1**
$\sqrt{39}$	1580	253	1	5	**2**	**3**
$\sqrt{50}$	799	113	0	8	**1**	**6**
$\sqrt{65}$	1169	145	0	9	**1**	**7**

$k=5$						
α	a	b	m	r	s	t
$\sqrt{80}$	197	22	1	**1**	**5**	4
$\sqrt{12}$	111	32	1	**2**	**3**	4
$\sqrt{82}$	1313	145	0	**8**	**1**	10
$\sqrt{101}$	1819	181	0	**9**	**1**	11
$\sqrt{80}$	653	73	1	4	**5**	**1**
$\sqrt{12}$	201	58	1	4	**3**	**2**
$\sqrt{82}$	1639	181	0	10	**1**	**8**
$\sqrt{101}$	2221	221	0	11	**1**	**9**

$k=6$						
α	a	b	m	r	s	t
$\sqrt{194}$	6421	461	3	**1**	**6**	5
$\sqrt{84}$	5105	557	1	**5**	**2**	7
$\sqrt{122}$	2441	221	0	**10**	**1**	12
$\sqrt{145}$	3191	265	0	**11**	**1**	13
$\sqrt{194}$	989	71	1	5	**6**	**1**
$\sqrt{84}$	7103	775	1	7	**2**	**5**
$\sqrt{122}$	2927	265	0	12	**1**	**10**
$\sqrt{145}$	3769	313	0	13	**1**	**11**

$k=7$						
α	a	b	m	r	s	t
$\sqrt{360}$	835	44	1	**1**	**7**	6
$\sqrt{48}$	215	31	1	**2**	**5**	3
$\sqrt{87}$	2136	229	1	**4**	**3**	5
$\sqrt{170}$	4081	313	0	**12**	**1**	14
$\sqrt{197}$	5123	365	0	**13**	**1**	15
$\sqrt{360}$	4345	229	1	6	**7**	**1**
$\sqrt{48}$	305	44	1	3	**5**	**2**
$\sqrt{87}$	2649	284	1	5	**3**	**4**
$\sqrt{170}$	4759	365	0	14	**1**	**12**
$\sqrt{197}$	5909	421	0	15	**1**	**13**

$k=8$						
α	a	b	m	r	s	t
$\sqrt{674}$	39799	1533	3	**1**	**8**	7
$\sqrt{90}$	1129	119	1	**3**	**4**	5
$\sqrt{147}$	16574	1367	1	**7**	**2**	9
$\sqrt{226}$	6329	421	0	**14**	**1**	16
$\sqrt{257}$	7711	481	0	**15**	**1**	17
$\sqrt{674}$	4751	183	1	7	**8**	**1**
$\sqrt{90}$	1831	193	1	5	**4**	**3**
$\sqrt{147}$	21254	1753	1	9	**2**	**7**
$\sqrt{226}$	7231	481	0	16	**1**	**14**
$\sqrt{257}$	8737	545	0	17	**1**	**15**

$k=9$						
α	a	b	m	r	s	t
$\sqrt{1088}$	2441	74	1	**1**	**9**	8
$\sqrt{105}$	4273	417	1	**5**	**3**	7
$\sqrt{290}$	9281	545	0	**16**	**1**	18
$\sqrt{325}$	11051	613	0	**17**	**1**	19
$\sqrt{1088}$	17449	529	1	8	**9**	**1**
$\sqrt{105}$	5933	579	1	7	**3**	**5**
$\sqrt{290}$	10439	613	0	18	**1**	**16**
$\sqrt{325}$	12349	685	0	19	**1**	**17**

$k=10$						
α	a	b	m	r	s	t
$\sqrt{1762}$	163917	3905	3	**1**	**10**	9
$\sqrt{228}$	41207	2729	1	**9**	**2**	11
$\sqrt{362}$	13033	685	0	**18**	**1**	20
$\sqrt{401}$	15239	761	0	**19**	**1**	21
$\sqrt{1762}$	15909	379	1	9	**10**	**1**
$\sqrt{228}$	50297	3331	1	11	**2**	**9**
$\sqrt{362}$	14479	761	0	20	**1**	**18**
$\sqrt{401}$	16841	841	0	21	**1**	**19**

$k=11$						
α	a	b	m	r	s	t
$\sqrt{2600}$	5711	112	1	**1**	**11**	10
$\sqrt{224}$	973	65	1	**2**	**7**	5
$\sqrt{240}$	2990	193	1	**3**	**5**	7
$\sqrt{442}$	17681	841	0	**20**	**1**	22
$\sqrt{485}$	20371	925	0	**21**	**1**	23
$\sqrt{2600}$	52061	1021	1	10	**11**	**1**
$\sqrt{224}$	2275	152	1	5	**7**	**2**
$\sqrt{240}$	6770	437	1	7	**5**	**3**
$\sqrt{442}$	19447	925	0	22	**1**	**20**
$\sqrt{485}$	22309	1013	0	23	**1**	**21**

$k=12$						
α	a	b	m	r	s	t
$\sqrt{3842}$	518743	8369	3	**1**	**12**	11
$\sqrt{235}$	7159	467	1	**5**	**4**	7
$\sqrt{27}$	1933	372	1	**7**	**3**	8
$\sqrt{327}$	86564	4787	1	**11**	**2**	13
$\sqrt{530}$	23321	1013	0	**22**	**1**	24
$\sqrt{577}$	26543	1105	0	**23**	**1**	25
$\sqrt{3842}$	42335	683	1	11	**12**	**1**
$\sqrt{235}$	9949	649	1	7	**4**	**5**
$\sqrt{27}$	2198	423	1	8	**3**	**7**
$\sqrt{327}$	102224	5653	1	13	**2**	**11**
$\sqrt{530}$	25439	1105	0	24	**1**	**22**
$\sqrt{577}$	28849	1201	0	25	**1**	**23**

For example, take the first row for $k=12$, i.e. $\alpha=\sqrt{3842}=[61,\overline{1,60,1,122}]$ and its rational approximation $\frac{518743}{8369}$, which satisfies $\left|\sqrt{3842}-\frac{518743}{8369}\right|<\frac{12}{8369^2}$. The convergents of $\sqrt{3842}$ are $\frac{61}{1}$, $\frac{62}{1}$, $\frac{3781}{61}$, $\frac{3843}{62}$, $\frac{472627}{7625}$, $\frac{476470}{7687}$, $\frac{29060827}{468845}$, ... The only representation of the fraction $\frac{518743}{8369}$ in the form $\frac{rp_{m+1}+sp_m}{rq_{m+1}+sq_m}$, $(r,s)\in R_{12}$ or $\frac{sp_{m+2}-tp_{m+1}}{sq_{m+2}-tq_{m+1}}$, $(s,t)\in T_{12}$ is $\frac{518743}{8369}=\frac{1\cdot 472627+12\cdot 3843}{1\cdot 7625+12\cdot 62}=\frac{1\cdot p_4+12\cdot p_3}{1\cdot q_4+12\cdot q_3}$, which shows that the pair $(1,12)$ cannot be omitted from the set R_{12}. □

3. Cases $r=1$, $s=1$ and $t=1$

The results from the previous section suggest that there are some patterns in pairs (r,s) and (s,t) which appear in representations $(a,b)=(rp_{m+1}+sp_m, rq_{m+1}+sq_m)$ and $(a,b)=(sp_{m+2}-tp_{m+1}, sq_{m+2}-tq_{m+1})$ of solutions of inequality (1.2). In particular, these patterns are easy to recognize for pairs of the form $(r,s)=(r,1)$ or $(1,s)$, and $(s,t)=(s,1)$ or $(1,t)$. In this section we will prove that the results on these pairs, already proved for $k\leqslant 12$, are valid in general. These facts will allow us to show that the inequality $rs<2k$ in Theorem 1.1 is sharp.

We will assume that k is a positive integer. From Theorem 1.1 it directly follows that among the pairs of the form $(r,1)$, only pairs where $r\leqslant 2k-1$ can appear. Similarly, for pairs $(1,t)$ we have $t\leqslant 2k-1$. On the other hand, from (2.3) and (2.5) it follows that for pairs $(1,s)$ we have $s\leqslant k$, and for pairs $(s,1)$ we have $s\leqslant k$. These results follow also from Theorem 1.2. We will show that all these pairs that do not contradict Theorem 1.2 can indeed appear.

Let $\alpha_m = [a_m; a_{m+1}, a_{m+2}, \ldots]$ and $\frac{1}{\beta_m} = \frac{q_{m-1}}{q_{m-2}} = [a_{m-1}, a_{m-2}, \ldots, a_1]$, with the convention that $\beta_1 = 0$. Then for $\frac{a}{b} = \frac{rp_{m+1}+sp_m}{rq_{m+1}+sq_m}$, we have

$$\begin{aligned} b^2\left|\alpha - \tfrac{a}{b}\right| &= b\left|(rq_{m+1}+sq_m)\tfrac{\alpha_{m+2}p_{m+1}+p_m}{\alpha_{m+2}q_{m+1}+q_m} - (rp_{m+1}+sp_m)\right| \\ &= \tfrac{|s\alpha_{m+2}-r|(rq_{m+1}+sq_m)}{\alpha_{m+2}q_{m+1}+q_m} = \tfrac{|s\alpha_{m+2}-r|(r+s\beta_{m+2})}{\alpha_{m+2}+\beta_{m+2}}. \end{aligned} \tag{3.1}$$

We start with the pairs of the form $(r, 1)$. Let us consider the number $\alpha = \sqrt{4k^2+1}$. Its continued fraction expansion has the form

$$\sqrt{4k^2+1} = [2k; \overline{4k}]$$

(see e.g. [8, p.297]). Take first $m = -1$, i.e. consider the rational number $\frac{a}{b}$ defined by

$$\frac{a}{b} = \frac{r \cdot p_0 + 1 \cdot p_{-1}}{r \cdot q_0 + 1 \cdot q_{-1}} = \frac{2rk+1}{r} = 2k + \frac{1}{r}.$$

Hence, $a = 2rk+1$ and $b = r$. We claim that for $r \leqslant 2k-1$, $\left|\alpha - \frac{a}{b}\right| < \frac{k}{b^2}$ holds. By (3.1), this is equivalent to $\left(1 - \frac{r}{\alpha_1}\right) r < k$. For $m \geqslant 1$ we have $\alpha_m = [4k, 4k, \ldots] < 4k + \frac{1}{4k}$. Thus, it suffices to check that $4kr^2 - (16k^2+1)r + 16k^3 + k > 0$, which is clearly satisfied for $r \leqslant 2k-1$. More precisely, this is satisfied for r less than $\frac{16k^2+1-\sqrt{16k^2+1}}{8k} > 2k - \frac{1}{2}$.

We can proceed similarly for $m \geqslant 0$. The only difference is that $4k < \frac{1}{\beta_{m+2}} = [4k, \ldots, 4k] < 4k + \frac{1}{4k}$. Hence, by (3.1), we obtain that it suffices to check that for $r \leqslant 2k-1$,

$$\left(4k + \frac{1}{4k} - r\right) \frac{r + \frac{1}{4k}}{4k + \frac{2}{4k+\frac{1}{4k}}} < k$$

holds. But this condition is equivalent to $(256k^4 + 16k^2)r^2 - (1024k^5 + 64k^3)r + (1024k^6 - 64k^4 - 32k^2 - 1) > 0$, which holds for r less than $2k - \frac{3}{4}$, so it certainly holds for $r \leqslant 2k-1$.

The same example $\alpha = \sqrt{4k^2+1}$ can be used to handle the pairs $(s, t) = (1, t)$. The relation (3.1) can be reformulated in terms of s and $t = sa_{m+2} - r$:

$$b^2\left|\alpha - \tfrac{a}{b}\right| = \left(t + \tfrac{s}{\alpha_{m+3}}\right)\left|s - \tfrac{t+\frac{s}{\alpha_{m+3}}}{\alpha_{m+2}+\beta_{m+2}}\right|. \tag{3.2}$$

Now, for $m = -1$ we are considering the rational number

$$\frac{a}{b} = \frac{s \cdot p_1 - t \cdot p_0}{s \cdot q_1 - t \cdot q_0} = \frac{8k^2 + 1 - 2tk}{4k - t} = 2k + \frac{1}{4k-t}.$$

By (3.2), the condition $\left|\alpha - \frac{a}{b}\right| < \frac{k}{b^2}$ leads to $16k^2t^2 - 64k^3t + 64k^4 - 12k^2 - 1 > 0$. Similarly, for $m \geqslant 0$, we obtain the condition $8k^2t^2 - (32k^3+2k)t + 32k^4 - 4k^2 - 1 > 0$. It is easy to see that both conditions are satisfied for $t \leqslant 2k-1$.

For pairs of the form $(1, s)$ and $(s, 1)$ we use α of the form $\alpha = \sqrt{x^2 - 1}$, where the integer x will be specified latter (if necessary). For $x \geqslant 2$, we have the following continued fraction expansion

$$\sqrt{x^2 - 1} = [x - 1; \overline{1, 2x - 2}]$$

(see e.g. [8, p.297]). Let us consider the pairs of the form $(r, s) = (1, s)$. We take $m = -1$ and define the rational number

$$\frac{a}{b} = \frac{1 \cdot p_0 + s \cdot p_{-1}}{1 \cdot q_0 + s \cdot q_{-1}} = \frac{x - 1 + s}{1}.$$

Hence, $a = x - 1 + s$ and $b = 1$, and for $s \leqslant k$,

$$\left|\alpha - \tfrac{a}{b}\right| < (x - 1 + s) - (x - 1) = s \leqslant \tfrac{k}{b^2}$$

holds. The same result for pairs $(r, s) = (1, k)$ holds also if $m \geqslant 1$ is odd and if x is sufficiently large. Indeed, from (3.1) we obtain the condition

$$\left(k\left(1 + \frac{1}{2x - 2}\right) - 1\right)\left(\frac{1 + \frac{k}{2x-2}}{1 + \frac{2}{2x-1}}\right) < k,$$

which is satisfied for $x \geqslant \frac{k^2-2k+5}{2}$.

Finally, consider the pairs of the form $(s, t) = (s, 1)$ for $s \leqslant k$. Take $m = -1$ and define the rational number

$$\frac{a}{b} = \frac{s \cdot p_1 - 1 \cdot p_0}{s \cdot q_1 - 1 \cdot q_0} = \frac{sx - x + 1}{s - 1} = x + \frac{1}{s - 1}.$$

Hence, $a = sx - x + 1$ and $b = s - 1$. We have $\sqrt{x^2 - 1} > \frac{p_2}{q_2} = x - \frac{1}{2x-1}$. Thus,

$$\left|\alpha - \tfrac{a}{b}\right| < \tfrac{1}{s-1} + \tfrac{1}{2x-1},$$

and we obtain the condition

$$\frac{1}{s - 1} + \frac{1}{2x - 1} < \frac{k}{(s - 1)^2}. \tag{3.3}$$

If we choose x to be greater than $\frac{k^2-2k+2}{2}$, then we have $\frac{1}{2x-1} < \frac{1}{(k-1)^2}$, while for $s \leqslant k$ the inequality $\frac{k}{(s-1)^2} - \frac{1}{s-1} \geqslant \frac{k}{(k-1)^2} - \frac{1}{k-1} = \frac{1}{(k-1)^2}$ holds, and we showed that for such x's the condition (3.3) is fulfilled.

Again, the analogous result for pairs $(s, t) = (k, 1)$ holds for all odd $m \geqslant 1$, but x has to be larger than in the case $m = -1$. Namely, the relation (3.2) yields the condition

$$\left(1 + \frac{k}{2x - 2}\right)\left(k - \frac{1}{1 + \frac{2}{2x-2}}\right) < k,$$

which is satisfied for $x \geqslant \frac{k^2-k+6}{2}$.

Our results for the pairs $(r,s) = (2k-1,1)$ and $(s,t) = (1,2k-1)$ (with $\alpha = \sqrt{4k^2+1}$) immediately imply the following result which shows that Theorem 1.1 is sharp.

Proposition 3.1. *For each $\varepsilon > 0$ there exist a positive integer k, a real number α and a rational number $\frac{a}{b}$, such that*

$$\left|\alpha - \frac{a}{b}\right| < \frac{k}{b^2},$$

and $\frac{a}{b}$ cannot be represented in the form $\frac{a}{b} = \frac{rp_{m+1} \pm sp_m}{rq_{m+1} \pm sq_m}$, for $m \geqslant -1$ and nonnegative integers r and s such that $rs < (2-\varepsilon)k$.

Proof. Take $k > \frac{1}{\varepsilon}$, $\alpha = \sqrt{4k^2+1}$ and e.g. $\frac{a}{b} = \frac{2k(2k-1)+1}{2k-1}$. Then $\left|\alpha - \frac{a}{b}\right| < \frac{k}{b^2}$. If $m = -1$, then $r = 2k-1$, $s = 1$, $t = 2k+1$, and thus $rs = 2k-1 > 2k-k\varepsilon = (2-\varepsilon)k$, while $st = 2k+1$. If $m \geqslant 0$, then from $s = -bp_{m+1} + aq_{m+1}$ it follows that $|s| \geqslant \left|\frac{a}{b} - \frac{p_1}{q_1}\right| bq_1 = 2k+1$, and therefore $|rs| \geqslant 2k+1$ and $|st| \geqslant 2k+1$. □

4. A Diophantine application

In [2], Dujella and Jadrijević considered the Thue inequality

$$\left|x^4 - 4cx^3y + (6c+2)\,x^2y^2 + 4cxy^3 + y^4\right| \leqslant 6c+4,$$

where $c \geqslant 3$ is an integer. In this section we will assume that $c \geqslant 5$, since the cases $c = 3$ and $c = 4$ require somewhat different details. Using the method of Tzanakis [9], they showed that solving the Thue equation $x^4 - 4cx^3y + (6c+2)\,x^2y^2 + 4cxy^3 + y^4 = \mu$, $\mu \in \mathbb{Z} \setminus \{0\}$, reduces to solving the system of Pellian equations

$$(2c+1)\,U^2 - 2cV^2 = \mu \tag{4.1}$$
$$(c-2)\,U^2 - cZ^2 = -2\mu, \tag{4.2}$$

where $U = x^2+y^2$, $V = x^2+xy-y^2$ and $Z = -x^2+4xy+y^2$. If suffices to find solutions of the system (4.1) and (4.2) which satisfy the condition $\gcd(U,V,Z) = 1$. Then $\gcd(U,V) = 1$, and $\gcd(U,Z) = 1$ or 2, since $4V^2+Z^2 = 5U^2$. It is clear that the solutions of the system (4.1) and (4.2) induce good rational approximations of the corresponding quadratic irrationals. More precisely, from [2, Lemma 4] we have the inequalities given in the following lemma.

Lemma 4.1. *Let $c \geqslant 5$ be an integer. All positive integer solutions (U,V,Z) of the system of Pellian equations (4.1) and (4.2) satisfy*

$$\left|\sqrt{\frac{2c+1}{2c}} - \frac{V}{U}\right| < \frac{2}{U^2} \tag{4.3}$$

$$\left|\sqrt{\frac{c-2}{c}}-\frac{Z}{U}\right| \quad < \quad \frac{6c+4}{U^2\sqrt{c\,(c-2)}} < \frac{9}{U^2}. \tag{4.4}$$

Using the result of Worley [12, Corollary, p. 206], in [2, Proposition 2] the authors proved that if μ is an integer such that $|\mu| \leqslant 6c+4$ and that the equation (4.1) has a solution in relatively prime integers U and V, then

$$\mu \in \{1,\, -2c,\, 2c+1,\, -6c+1,\, 6c+4\}.$$

Analysing the system (4.1) and (4.2), and using the properties of convergents of $\sqrt{\frac{2c+1}{2c}}$, they were able to show that the system has no solutions for $\mu = -2c, 2c+1, -6c+1$. Applying results from the previous sections to the equation (4.2), we will present here a new proof of that result, based on the precise information on μ's for which (4.2) has a solution in integers U and Z such that $\gcd(U,Z) \in \{1,2\}$.

The simple continued fraction expansion of a quadratic irrational $\alpha = \frac{e+\sqrt{d}}{f}$ is periodic. This expansion can be obtained using the following algorithm. Multiplying the numerator and the denominator by f, if necessary, we may assume that $f|(d-e^2)$. Let $s_0 = e$, $t_0 = f$ and

$$a_n = \left\lfloor \frac{s_n+\sqrt{d}}{t_n} \right\rfloor, \quad s_{n+1} = a_n t_n - s_n, \quad t_{n+1} = \frac{d-s_{n+1}^2}{t_n} \quad \text{for } n \geqslant 0 \tag{4.5}$$

(see [6, Chapter 7.7]). If $(s_j, t_j) = (s_k, t_k)$ for $j < k$, then

$$\alpha = [a_0; \ldots, a_{j-1}, \overline{a_j, \ldots, a_{k-1}}].$$

Applying this algorithm to $\sqrt{\frac{c-2}{c}} = \frac{\sqrt{c(c-2)}}{c}$, we find that

$$\sqrt{\frac{c-2}{c}} = [0; 1, \overline{c-2, 2}].$$

According to our results (Proposition 2.2 for $k=9$), applied to $\alpha = \sqrt{\frac{c-2}{c}}$, all solutions of (4.2) have the form $Z/U = (rp_{m+1} + sp_m)/(rq_{m+1} + sq_m)$ an index $m \geqslant -1$ and integers r and s. For the determination of the corresponding μ's, we use the following result (see [2, Lemma 1]):

Lemma 4.2. *Let $\alpha\beta$ be a positive integer which is not a perfect square, and let p_m/q_m denotes the mth convergent of continued fraction expansion of $\sqrt{\frac{\alpha}{\beta}}$. Let the sequences (s_m) and (t_m) be defined by (4.5) for the quadratic irrational $\frac{\sqrt{\alpha\beta}}{\beta}$. Then*

$$\begin{aligned} &\alpha(rq_{m+1} + sq_m)^2 - \beta(rp_{m+1} + sp_m)^2 \\ &= (-1)^m(s^2 t_{m+1} + 2rss_{m+2} - r^2 t_{m+2}). \end{aligned} \tag{4.6}$$

Since the period of the continued fraction expansion of $\sqrt{\frac{c-2}{c}}$ is equal to 2, according to Lemma 4.2, we have to consider only the fractions $\frac{rp_{m+1}+sp_m}{rq_{m+1}+sq_m}$ for $m=1$ and $m=2$. By checking all possibilities, we obtain the following result.

Proposition 4.3. *Let μ be an integer such that $|\mu| \leqslant 6c+4$ and that the equation*

$$(c-2)U^2 - cZ^2 = -2\mu$$

has a solution in integers U and Z such that $\gcd(U,Z) = 1$ or 2.

(i) *If $c \geqslant 15$ is odd, then*

$$\mu \in M_1 = \{1, 4, 2c, 4c+1, 6c+4, -2c+4, -4c+9, -6c+16\}.$$

Furthermore, if $c = 5, 11, 13$, then $\mu \in M_1 \cup \{-8c+25\}$; if $c = 9$, then $\mu \in M_1 \cup \{-8c+25, -10c+36\}$; if $c = 7$, then $\mu \in M_1 \cup \{-8c+25, -10c+36, -12c+49\}$.

(ii) *Let $M = M_1 \cup M_2$, where*

$$M_2 = \left\{-\frac{11}{2}c+36, -\frac{9}{2}c+25, -\frac{7}{2}c+16, -\frac{5}{2}c+9, -\frac{3}{2}c+4, -\frac{1}{2}c+1, \frac{1}{2}c, \frac{3}{2}c+1, \frac{5}{2}c+4, \frac{7}{2}c+9\right\}.$$

If $c \geqslant 108$ is even, then $\mu \in M \cup \left\{\frac{9}{2}c+16, \frac{11}{2}c+25\right\}$.

For even c with $6 \leqslant c \leqslant 106$, we have $\mu \in M \cup M^{(c)}$, where $M^{(c)}$ can be given explicitly, as in the case (i). E.g.

$$M^{(6)} = \left\{-\frac{21}{2}c+25, -10c+36, -8c+25, -\frac{15}{2}c+16\right\}.$$

Comparing the set $\{1, -2c, 2c+1, -6c+1, 6c+4\}$ from [2, Proposition 2] with the sets appearing in Proposition 4.3, we obtain the desired conclusion.

Corollary 4.4. *Let $c \geqslant 5$ be an integer. If the system (4.1) and (4.2) has a solution with $|\mu| \leqslant 6c+4$ in integers U, V and Z such that $\gcd(U,V,Z) = 1$, then $\mu = 1$ or $\mu = 6c+4$.*

Acknowledgements. The authors would like to thank the referee for valuable remarks and suggestions.

References

[1] Dujella, A., Continued fractions and RSA with small secret exponents, *Tatra Mt. Math. Publ.*, 29 (2004) 101–112.

[2] Dujella, A., Jadrijević, B., A family of quartic Thue inequalities, *Acta Arith.*, 111 (2004) 61–76.

[3] Fatou, P., Sur l'approximation des incommenurables et les series trigonometriques, *C. R. Acad. Sci. (Paris)*, 139 (1904) 1019–1021.

[4] Koksma, J.F., On continued fraction, *Simon Stevin*, 29 (1951/52) 96–102.

[5] Lang, S., Introduction to Diophantine Approximations, *Addison-Wesley*, Reading, 1966.

[6] Niven, I., Zuckerman, H.S., Montgomery, H.L., An Introduction to the Theory of Numbers, *John Wiley*, New York, 1991.

[7] Osgood, C.F., Luca, F., Walsh, P.G., Diophantine approximations and a problem from the 1988 IMO, *Rocky Mountain J. Math.*, 36 (2006) 637–648.

[8] Sierpiński, W., Elementary Theory of Numbers, PWN, Warszawa; North-Holland, Amsterdam, 1987.

[9] Tzanakis, N., Explicit solution of a class of quartic Thue equations, *Acta Arith.* 64 (1993) 271–283.

[10] Verheul, E.R., van Tilborg, H.C.A., Cryptanalysis of 'less short' RSA secret exponents, *Appl. Algebra Engrg. Comm. Computing*, 8 (1997) 425–435.

[11] Wiener, M.J., Cryptanalysis of short RSA secret exponents, *IEEE Trans. Inform. Theory*, 36 (1990) 553–558.

[12] Worley, R.T., Estimating $|\alpha - p/q|$, *J. Austral. Math. Soc. Ser. A*, 31 (1981) 202–206.

Andrej Dujella
Department of Mathematics, University of Zagreb
Bijenička cesta 30, 10000 Zagreb, Croatia
e-mail: duje@math.hr

Bernadin Ibrahimpašić
Pedagogical Faculty, University of Bihać
Džanića mahala 36, 77000 Bihać, Bosnia and Herzegovina
e-mail: bernadin@bih.net.ba

Annales Mathematicae et Informaticae
***35** (2008) pp. 75–94*
http://www.ektf.hu/ami

Simplifying the propositional satisfiability problem by sub-model propagation*

Gábor Kusper, Lajos Csőke, Gergely Kovásznai

Institute of Mathematics and Informatics
Eszterházy Károly College, Eger, Hungary

Submitted 30 September 2008; Accepted 8 December 2008

Abstract

We describes cases when we can simplify a general SAT problem instance by sub-model propagation. Assume that we test our input clause set whether it is blocked or not, because we know that a blocked clause set can be solved in polynomial time. If the input clause set is not blocked, but some clauses are blocked, then what can we do? Can we use the blocked clauses to simplify the clause set? The Blocked Clear Clause Rule and the Independent Blocked Clause Rule describe cases when the answer is yes. The other two independent clause rules, the Independent Nondecisive- and Independent Strongly Nondecisive Clause Rules describe cases when we can use nondecisive and strongly nondecisive clauses to simplify a general SAT problem instance.

Keywords: SAT, blocked clause, nondecisive clause

MSC: 03-04

1. Introduction

Propositional Satisfiability is the problem of determining, for a formula of the propositional calculus, if there is an assignment of truth values to its variables for which that formula evaluates the true. By SAT we mean the problem of propositional satisfiability for formulae in conjunctive normal form (CNF).

SAT is the first, and one of the simplest, of the many problems which have been shown to be NP-complete [7]. It is dual of propositional theorem proving, and many practical NP-hard problems may be transformed efficiently to SAT. Thus, a good SAT algorithm would likely have considerable utility. It seems improbable that a polynomial time algorithm will be found for the general SAT problem but we know

*Partially supported by TéT 2006/A-16.

that there are restricted SAT problems that are solvable in polynomial time. So a "good" SAT algorithm should check first the input SAT instance whether it is an instance of such a restricted SAT problem or can be simplified by a preprocess step. In this paper we introduce some possible simplification techniques. We list some polynomial time solvable restricted SAT problems:

1. The restriction of SAT to instances where all clauses have length k is denoted by k-SAT. Of special interest are *2-SAT* and *3-SAT*: 3 is the smallest value of k for which k-SAT is **NP**-complete, while 2-SAT is solvable in linear time [10, 1].

2. *Horn SAT* is the restriction to instances where each clause has at most one positive literal. Horn SAT is solvable in linear time [9, 19], as are a number of generalizations such as *renamable Horn SAT* [2], *extended Horn SAT* [5] and *q-Horn SAT* [3, 4].

3. The hierarchy of *tractable* satisfiability problems [8], which is based on Horn SAT and 2-SAT, is solvable in polynomial time. An instance on the k-th level of the hierarchy is solvable in $O(nk + 1)$ time.

4. *Nested SAT*, in which there is a linear ordering on the variables and no two clauses overlap with respect to the interval defined by the variables they contain [12].

5. SAT in which no variable appears more than twice. All such problems are satisfiable if they contain no unit clauses [20].

6. *r,r-SAT*, where r,s-SAT is the class of problems in which every clause has exactly r literals and every variable has at most s occurrences. All r,r-SAT problems are satisfiable in polynomial time [20].

7. A formula is *SLUR* (Single Lookahead Unit Resolution) *solvable* if, for all possible sequences of selected variables, algorithm SLUR does not give up. Algorithm SLUR is a nondeterministic algorithm based on unit propagation. It eventually gives up the search if it starts with, or creates, an unsatisfiable formula with no unit clauses. The class of SLUR solvable formulae was developed as a generalization including Horn SAT, renamable Horn SAT, extended Horn SAT, and the class of CC-balanced formulae [18].

8. *Resolution-Free* SAT Problem, where every resolution results in a tautologous clause, is solvable in linear time [16].

8. *Blocked* SAT Problem, where every clause is blocked, is solvable in polynomial time [13, 14, 17].

In this paper we describes cases when we can simplify a general SAT problem instance by sub-model propagation, which means hyper-unit propagating [15, 16] a sub-model [17]. Assume that we test our input clause set whether it is blocked or not, because we know [17] that a blocked clause set can be solved in polynomial time. If the input clause set is not blocked, but some clauses are blocked, then what can we do? Can we use the blocked clauses to simplify the clause set? The Blocked Clear Clause Rule and the Independent Blocked Clause Rule describe cases when the answer is yes.

The other two independent clause rules, the Independent Nondecisive- and Independent Strongly Nondecisive Clause Rules describe cases when we can use nonde-

cisive and strongly nondecisive clauses to simplify a general SAT problem instance.

The notion of blocked [13, 14] and nondecisive clause [11] was introduced by O. Kullmann and A. V. Gelder. They showed that a blocked or nondecisive clause can be added or deleted from a clause set without changing its satisfiability.

Intuitively a blocked clause has a liter on which every resolution in the clause set is tautology. A nondecisive clause has a literal on which every resolution in the clause set is either tautology or subsumed. We also use the notion of strongly nondecisive clause, which has a liter on which every resolution in the clause set is either tautology or entailed. We also use very frequently the notion of clear clause. A clause is clear if every variable which occurs in the clause set occurs also in this clause either positively or negatively. Note, that clear clauses are called also total or full clauses in the literature.

The Blocked Clear Clause Rule describes two cases. The two cases have a common property: the input clause set contains a blocked clear clause. In the first case the input clause set is a subset of CC, in the second case the blocked clear clause is not subsumed. In both cases the sub-model generated from the blocked clear clause and from one of its blocked literals is a model for the input clause set.

In both cases we need in the worst-case $O(n^2m^3)$ time to decide whether the input clause set fulfills the requirements of the Blocked Clear Clause Rule. We need $O(n^2m^3)$ time, because we have to check blocked-ness in both two cases, which is an $O(n^2m^2)$ time method, and not subsumed-ness in the second case, which is an $O(m)$ time method.

The Independent Blocked Clause Rule is a generalization of the Blocked Clear Clause Rule. We can apply it if we have a blocked clause and it subsumes a clear clause that is it not subsumed by any other clause from the clause set, i.e., the blocked clause is independent. In this case the sub-model generated from the independent blocked clause and from one of its blocked literals is a partial model, i.e., we can simplify the input clause set by propagating this sub-model.

Note that if we know the subsumed clear clause which is not subsumed by any other clause from the input clause set then we know the whole model. This applies for the other independent clause rules.

We need in the worst-case $O(2^n n^2m^3)$ time to decide whether the input clause set fulfills the requirements of the Independent Blocked Clause Rule. We need $O(2^n n^2m^3)$ time, because we have to check blocked-ness, which is an $O(n^2m^2)$ time method, and independent-ness, which is an $O(2^n m)$ time method.

The Independent Nondecisive Clause Rule is a generalization of the Independent Blocked Clause Rule. We can apply it if we have a independent nondecisive clause. In this case the sub-model generated from it and from one of its nondecisive literals is a partial model, i.e., we can simplify the input clause set by propagating this sub-model.

We need in the worst-case $O(2^n nm^4)$ time to decide whether the input clause set fulfills the requirements of the Independent Nondecisive Clause Rule. We need $O(2^n nm^4)$ time, because we have to check nondecisive-ness, which is an $Max\{O(n^2m^2), O(nm^3)\}$ time method, and independent-ness, which is an $O(2^n m)$

time method. We assume that $nm^3 > n^2m^2$.

The Independent Strongly Nondecisive Clause Rule is a generalization of the Independent Nondecisive Clause Rule. We can apply it if we have a independent strongly nondecisive clause. In this case the sub-model generated from it and from one of its strongly nondecisive literals is a partial model, i.e., we can simplify the input clause set by propagating this sub-model.

We need in the worst-case $O(2^{n+1}m)$ time to decide whether the input clause set fulfills the requirements of the Independent Strongly Nondecisive Clause Rule. We need $O(2^{n+1}m)$ time, because we have to check strongly nondecisive-ness, which is an $O(n^2)$ time method, and independent-ness, which is an $O(2^n m)$ time method.

Since the independent clause test is too expensive (it is exponential) we introduce some heuristics which can guess which clause might be independent. Furthermore, we introduce an algorithm which might find strongly nondecisive clauses in $O(n^3m^2)$ time.

2. Definitions

Set of variables, literals

Let V be a finite *set of Boolean variables*. The *negation of a variable* v is denoted by $\overline{v}$. Given a set U, we denote $\overline{U} := \{\overline{u} \mid u \in U\}$ and call the *negation of the set* U.

Literals are the members of the set $W := V \cup \overline{V}$. *Positive literals* are the members of the set V. *Negative literals* are their negations. If w denotes a negative literal $\overline{v}$, then $\overline{w}$ denotes the positive literal v.

Clause, clause set, assignment, assignment set

Clauses and *assignments* are finite sets of literals that do not contain simultaneously any literal together with its negation.

A clause is interpreted as disjunction of its literals. An assignment is interpreted as conjunction of its literals. Informally speaking, if an assignment A contains a literal v, it means that v has the value $True \in A$. A *clause set* or *formula* (formula in CNF form) is a finite set of clauses. A clause set is interpreted as conjunction of its clauses. If C is a clause, then $\overline{C}$ is an assignment. If A is an assignment, then $\overline{A}$ is a clause. The empty clause is interpreted as False. The empty assignment is interpreted as True. The empty clause set is interpreted as True.

The empty set is denoted by $\emptyset$. The *length* of a set U is its cardinality, denoted by $|U|$. The natural number n is the *number of variables*, i.e., $n := |V|$.

Cardinality, *k*-clause, clear clause, CC

If C is a clause and $|C| = k$, then we say that C is a *k-clause*. Special cases are *unit clauses* or *units* which are 1-clauses, and *clear* or *total clauses* which are

n-clauses. Note that any unit clause is at the same time a clause and an assignment.

In this paper we prefer the name clear clause instead of total or full clause. Although, total clause is used in the literature, in our point of view the name clear clause is more intuitive.

The clause set CC is the set of all clear clauses.

Subsumption, entailed-ness, independent-ness

The clause C *subsumes* the clause B iff C is a subset of B. The interpretation of the notion of subsumption is logical consequence, i.e., B is a logical consequence of C.

We say that a clause C is *subsumed by the clause set* S, denoted by $C \supseteq\in S$, iff there is a clause in S which subsumes it. We say that a clause C is *entailed by the clause set* S, denoted by $C \supseteq\in_{CC} S$, iff for any clear clause, which is subsumed by C, there is a clause in S which subsumes that clear clause.

The interpretation of the notion of subsumed and entailed is the same, logical consequence, i.e., C is a logical consequence of S. Note that if a clause is subsumed by a clause set then it is entailed, but not the other way around. Furthermore, if a clear clause is subsumed by a clause set then it is entailed and the other way around.

$$C \supseteq\in S :\Longleftrightarrow \mathit{Clause}(C) \wedge \mathit{ClauseSet}(S) \wedge \exists[B \in S]B \subseteq C.$$

$$C \supseteq\in_{CC} S :\Longleftrightarrow \mathit{Clause}(C) \wedge \mathit{ClauseSet}(S) \wedge \forall[D \in CC][C \subseteq D]\exists[B \in S]B \subseteq D.$$

We shall explain the intuition behind the notation $\supseteq\in$. If we rewrite its definition and leave out the "not interesting" parts (written in brackets) then we obtain this notation:

$$\exists[B \in S]B \subseteq C \iff (\exists[B])C \supseteq (B \wedge B) \in S \iff C \supseteq\in S.$$

We say that a clause C is *independent in clause set* S iff it is not entailed by S.

Clause difference, resolution

We introduce the notion of *clause difference*. We say that two clauses *differ in* some variables iff these variables occur in both clauses but as different literals. If A and B are clauses then the clause difference of them, denoted by $\mathrm{diff}(A, B)$, is

$$\mathrm{diff}(A, B) := A \cap \overline{B}.$$

If $\mathrm{diff}(A, B) \neq \emptyset$ then we say that A *differs from* B. Note that $\mathrm{diff}(A, B) = \overline{\mathrm{diff}(B, A)}$.

We say that *resolution can be performed* on two clauses iff they differ only in one variable. Note that this is not the usual notion of resolution, because we allow resolution only if it results in a non-tautologous resolvent. For example resolution cannot be performed on $\{v, w\}$ and $\{\overline{v}, \overline{w}\}$ but can be performed on $\{v, w\}$ and

$\{\overline{v}, z\}$. If resolution can be performed on two clauses, say A and B, then the *resolvent*, denoted by $\mathrm{Res}(A, B)$, is their union excluding the variable they differ in:

$$\mathrm{Res}(A, B) := (A \cup B) \setminus (\mathrm{diff}(A, B) \cup \mathrm{diff}(B, A)).$$

Note that if we interpret $\mathrm{Res}(A, B)$ as a logical formula then it is a logical consequence of the clauses A and B.

Pure literal, blocked- literal, clause, clause set

We say that a literal $c \in C$ is *blocked in* the clause C and in the clause set S iff for each clause B *in*S which contains $\overline{c}$ we have that there is a literal $b \in B$ such that $b \neq \overline{c}$ and $\overline{b} \in C$. A *clause is blocked in a clause set* iff it contains a blocked literal. A *clause set is blocked* iff all clauses are blocked in it. We denote these notions by $Blck(c, C, S)$, $Blck(C, S)$ and $Blck(S)$, respectively.

Note that if literal $c \in C$ is blocked in C, S then for all $B \in S, \overline{c} \in B$ we have that resolution cannot be performed on C and B. This means that this clause is "blocked" against resolution.

We say that a literal is *pure* in a clause set if its negation does not occur in the clause set. Note that pure literals are blocked.

(Weakly/strongly) nondecisive- literal, clause, clause set

We define formally the notion of *weakly nondecisive* literal, clause and clause set. We denote these notions by $WnD(c, C, S)$, $WnD(C, S)$ and $WnD(S)$, respectively.

$$WnD(c, C, S) : \Longleftrightarrow \forall[B \in S][\overline{c} \in B](\exists[b \in B][b \neq \overline{c}]\overline{b} \in C \vee \mathrm{Res}(C, B) \supseteq \in S).$$

$$WnD(C, S) : \Longleftrightarrow \exists[c \in C]WnD(c, C, S).$$

$$WnD(S) : \Longleftrightarrow \forall[C \in S]WnD(C, S).$$

We define formally the notion of*nondecisive* literal, clause and clause set. We denote these notions by $NonD(c, C, S)$, $NonD(C, S)$ and $NonD(S)$, respectively.

$$NonD(c, C, S) : \Longleftrightarrow$$

$$\forall[B \in S][\overline{c} \in B](\exists[b \in B][b \neq \overline{c}]\overline{b} \in C \vee \mathrm{Res}(C, B) \cup \{c\} \supseteq \in S \setminus \{C\}).$$

$$NonD(C, S) : \Longleftrightarrow \exists[c \in C]NonD(c, C, S).$$

$$NonD(S) : \Longleftrightarrow \forall[C \in S]NonD(C, S).$$

We define formally the notion of *strongly nondecisive* literal, clause and clause set. We denote these notions by $SND(c, C, S)$, $SND(C, S)$ and $SND(S)$, respectively.

$$SND(c, C, S) : \Longleftrightarrow$$

$$\forall[B \in S][\overline{c} \in B](\exists[b \in B][b \neq \overline{c}]\overline{b} \in C \vee \mathrm{Res}(C, B) \cup \{c\} \supseteq \in_{CC} S \setminus \{C\}).$$

$$SND(C, S) : \Longleftrightarrow \exists[c \in C]SND(c, C, S).$$

$$SND(S) : \Longleftrightarrow \forall[C \in S]SND(C, S).$$

Resolution-mate, sub-model

If C is a clause and c is a literal in C then the *resolution-mate* of clause C by literal c, denoted by $\mathrm{rm}(C, c)$, is

$$\mathrm{rm}(C, c) := (C \cup \{\overline{c}\}) \setminus \{c\}.$$

Note that resolution can be always performed on C and $\mathrm{rm}(C, c)$, and

$$\mathrm{Res}(C, \mathrm{rm}(C, c)) = C \setminus \{c\}.$$

This means that we obtain a shorter clause.

The *sub-model* generated from the clause C and from the literal c, denoted by $\mathrm{sm}(C, c)$, is

$$\mathrm{sm}(C, c) := \overline{\mathrm{rm}(C, c)}.$$

We say that C and c are the *generator* of $\mathrm{sm}(C, c)$. The name "sub-model" comes from the observation that in a resolution-free clause set an assignment created from one of the shortest clauses in this way is a part of a model [16], i.e., a sub-model.

Note that $\mathrm{rm}(C, c)$ is a clause but $\mathrm{sm}(C, c)$ is an assignment.

The sub-model $\mathrm{sm}(C, c)$ is a special assignment which always satisfies clause C, since it sets literal c to be True.

Model, (un)satisfiable

An assignment M is a *model for a clause set* S iff for all $C \in S$ we have $M \cap C \neq \emptyset$.

A clause set is *satisfiable* iff there is a model for it. A clause set is *unsatisfiable* iff it is not satisfiable. A clause set is *trivially satisfiable* iff it is empty and it is *trivially unsatisfiable* if it contains the empty clause.

3. The Blocked Clear Clause Rule

In this section we introduce the Blocked Clear Clause Rule, a generalization of the Clear Clause Rule. This rule is introduced by the author.

Assume we test our input clause set whether it is blocked or not, because we know [17] that a blocked clause set can be solved in polynomial time. If the input clause set is not blocked, but some clauses are blocked, then what can we do? Can we use the blocked clauses to simplify the clause set? If it contains a not subsumed blocked clear clause, we can. This is what the Blocked Clear Clause Rule states.

It has two variants. The first one states that if a clause set contains only clear clauses and one of them is blocked then the sub-model generated from this blocked clause and from one of its blocked literal is a model. This is a very rare case, but since we can construct for each clause set the equivalent clear clause set, this rule plays an important role.

The second one states that if a clause set contains a not subsumed blocked clear clause then the sub-model generated from it and from one of its blocked literals is a model. This case is still a very rare one, but might occur more frequently as the first variant.

Lemma 3.1 (Blocked Clear Clause Rule). *Let S be a clause set. Let $C \in S$ be a blocked and clear clause. Let $a \in C$ be a blocked literal C, S.*
(a) *If S is a subset of CC, then $sm(C, a)$ is a model for S.*
(b) *If C is not subsumed by $S \setminus \{C\}$, then $sm(C, a)$ is a model for S.*

Proof. (a) To show this, by definition of model, it suffices to show that for an arbitrary but fixed $B \in S$ we have that $B \cap \mathrm{sm}(C, a)$ is not empty. Since S is a subset of CC we know that B is a clear clause. Hence, there are two cases, either $a \in B$ or $\overline{a} \in B$.

In case $a \in B$ we have, by definition of sub-model, that $a \in \mathrm{sm}(C, a)$. Hence, $B \cap \mathrm{sm}(C, a)$ is not empty.

In case $\overline{a} \in B$, since $a \in C$ is blocked in C, S we know, by definition of blocked literal, that for some $b \in B$ we have $b \neq \overline{a}$ and $\overline{b} \in C$. From this, by definition of sub-model, we know that $b \in \mathrm{sm}(C, a)$. Hence, $B \cap \mathrm{sm}(C, a)$ is not empty.

Hence, if S is a subset of CC, then $\mathrm{sm}(C, a)$ is a model for S.

(b) To show this, by definition of model, it suffices to show that for an arbitrary but fixed $B \in S$ we have that $B \cap \mathrm{sm}(C, a)$ is not empty. Since C is not subsumed by $S \setminus \{C\}$ we know, by definition of subsumption, that $B \not\subseteq C$. From this, since C is a clear clause we know that for some $b \in B$ we have $\overline{b} \in C$. There are two cases, either $\overline{b} = a$ or $\overline{b} \neq a$.

In the first case we have $\overline{b} = a$, i.e., $\overline{a} \in B$. From this since $a \in C$ is blocked in C, S we know, by definition of blocked literal, that for some $d \in B$ we have that $d \neq \overline{a}$ and $\overline{d} \in C$. From this, by definition of sub-model, we know that $d \in \mathrm{sm}(C, a)$. Hence, $B \cap \mathrm{sm}(C, a)$ is not empty.

In the second case we have $\overline{b} \neq a$. From this and from $\overline{b} \in C$ we know, by definition of sub-model, that $b \in \mathrm{sm}(C, a)$. Hence, $B \cap \mathrm{sm}(C, a)$ is not empty.

Hence, If C is not subsumed by $S \setminus \{C\}$, then $\mathrm{sm}(C, a)$ is a model for S. □

An alternative proof idea is that we say that it suffices to show that the resolution-mate of C ($\mathrm{rm}(C, a)$) is not subsumed by S. Then we know, by Clear Clause Rule, that its negation ($\mathrm{sm}(C, a)$) is a model.

This alternative proof idea shows in which sense say we that the Blocked Clear Clause Rule is a generalization of the Clear Clause Rule.

This rule is the base of the independent clause rules. Therefore, it is very important for us.

4. The Independent Blocked Clause Rule

In this section we introduce the Independent Blocked Clause Rule, a generalization of the Blocked Clear Clause Rule. This rule is introduced by the author.

The Independent Blocked Clause Rule states that if a clause set contains an independent blocked clause, then it is satisfiable and a sub-model generated from this clause and from one of its blocked literals is a partial model, i.e., we can simplify the clause set by propagating this sub-model. These requirements are fulfilled quite often by real or benchmark problems, but checking independent-ness is expensive.

We know that a clause $A \in S$ is independent in the clause set $S \setminus \{A\}$ if it is not entailed by $S \setminus \{A\}$. The formal definition is the following:

$$A \text{ independent in } S : \Longleftrightarrow \exists[C \in CC][A \subseteq C]\forall[B \in S][B \neq A]B \not\subseteq C.$$

The following algorithm checks whether the input clause is independent or not in the input clause. If it is independent, then it returns a clear clause subsumed by the input clause but not subsumed by any other clause from the input clause set. Otherwise, it returns the empty clause. In the worst-case it uses $O(2^n m)$ time, because it follows the definition of independent, and there we have two quantifiers, one on CC which has 2^n elements, the other on the input clause set, which has m elements.

Independent clause test

```
1  function IsIndependent(S : clause set, A : clause) : clause
2  begin
3    for each C ∈ CC, A ⊆ C do
4      B_notsubsumes_C := True;
5      for each B ∈ S, B ≠ A while B_notsubsumes_C is True do
6        if (B ⊆ C) then B_notsubsumes_C := False;
7      od
8      if (B_notsubsumes_C) then return C;
9        // In this case we found a suitable C, we return it.
10   od
11   return ∅;
12     // In this case we found no suitable clause.
13     // Therefore, we return the empty clause.
14 end
```

One can see that the independent clause test is very expensive (exponential). We will discuss later how can we get around this problem by suitable heuristics.

Lemma 4.1 (Independent Blocked Clause Rule). *Let S be a clause set. Let $A \in S$ be blocked in S and independent in $S \setminus \{A\}$. Let $a \in A$ be a blocked literal in A, S. Then there is a model M for S such that $sm(A, a) \subseteq M$.*

Proof. We know that A is independent in $S \setminus \{A\}$. Hence, by definition of independent, we know that there is a clear clause C that is subsumed by A and not subsumed by any other clause in S. Since $A \subseteq C$ we know that $\mathrm{sm}(A, a) \subseteq \mathrm{sm}(C, a)$. Hence, it suffices to show that $\mathrm{sm}(C, a)$ is a model for S. To show this, by definition of model, it suffices to show that for an arbitrary but fixed $B \in S$ we have that $B \cap \mathrm{sm}(C, a)$ is not empty. The remaining part of the proof is the same as the proof of the (b) variant of the Blocked Clear Clause Rule.

Hence, $B \cap \mathrm{sm}(C, a)$ is not empty. Hence, there is a model M for S such that $\mathrm{sm}(A, a) \subseteq M$. □

This proof is traced back to the proof of Blocked Clear Clause Rule. We can do this because we know that there is a clear clause which is blocked and not entailed by $S \setminus \{A\}$. We know that for clear clauses the notion of subsumed and entailed are the same.

The proof of this lemma shows that if we perform an independent clause check and we find a clear clause which is subsumed by only one clause, then we know the whole model ($\mathrm{sm}(C, a)$) and not only a part of the model ($\mathrm{sm}(A, a)$). But usually we do not want to perform expensive independent-ness checks. How can we get around this problem? The solution is a heuristic which tells us which blocked clause could be independent.

Such a heuristic could be for instance the selection of the shortest blocked clause. The shortest clause subsumes the largest number of clear clauses. Therefore, it has a good chance to be independent, but there is no guarantee for it. We give more details about heuristics after the discussion of the simplifying rules.

5. The Independent Nondecisive Clause Rule

In this section we introduce the Independent Nondecisive Clause Rule, a generalization of the Independent Blocked Clause Rule. This rule is introduced by the author.

The Independent Nondecisive Clause Rule states that if a clause set contains an independent nondecisive clause, then it is satisfiable and a sub-model generated from this clause and from one of its nondecisive literals is a partial model, i.e., we can simplify the clause set by propagating this sub-model. These requirements are fulfilled quite often by real or benchmark problems, but checking independent-ness is expensive.

Lemma 5.1 (Independent Nondecisive Clause Rule). *Let S be a clause set. Let $A \in S$ be nondecisive in S and independent in $S \setminus \{A\}$. Let $a \in A$ be a nondecisive literal in A, S. Then there is a model M for S such that $sm(A, a) \subseteq M$.*

Proof. We know that A is independent in $S \setminus \{A\}$. Hence, by definition of independent, we know that there is a clear clause C that is subsumed by A and not subsumed by any other clause in S. Since $A \subseteq C$ we know that $\mathrm{sm}(A, a) \subseteq \mathrm{sm}(C, a)$. Hence, it suffices to show that $\mathrm{sm}(C, a)$ is a model for S. To show this, by definition

of model, it suffices to show that for an arbitrary but fixed $B \in S$ we have that $B \cap \mathrm{sm}(C, a)$ is not empty. There are three cases: either (a) $a \in B$ or (b) $\overline{a} \in B$ or (c) $a \notin B$ and $\overline{a} \notin B$.

In case (a) we have $a \in B$. From this and from the definition of sub-model we know that $a \in B \cap \mathrm{sm}(C, a)$.

In case (b) we have $\overline{a} \in B$. From this and from $a \in A$ is nondecisive in A, S, by definition of nondecisive literal, we know that either there is a literal $b \in B$ which has $b \neq \overline{a}$ and $\overline{b} \in A$ or there is a clause $D \in S, D \neq A$ which has $D \subseteq A \cup B\{\overline{a}\}$.

In the first case we know, by definition of sub-model, that $b \in \mathrm{sm}(A, a)$.

In the second case since C is independent in $S \setminus \{A\}$, by definition of independent, we know that D does not subsume C, i.e., for some $d \in D$ we have $d \notin C$. From this and from $A \subseteq C$ and from $D \subseteq A \cup B\{\overline{a}\}$ we can show that $d \notin A$, $d \in B$ and $d \neq \overline{a}$. From $d \notin C$ we know, by definition of clear clause, that $\overline{d} \in C$. Hence, by definition of sub-model, $d \in B \cap \mathrm{sm}(C, a)$.

In case (c) we have $a \notin B$ and $\overline{a} \notin B$. Since C is not subsumed by $S \setminus \{A\}$ we know, by definition of subsumption, that $B \not\subseteq C$. From this, since C is a clear clause we know that for some $b \in B$ we have $\overline{b} \in C$. There are two cases, either $\overline{b} = a$ or $\overline{b} \neq a$.

In the first case we have $\overline{b} = a$, i.e., $\overline{a} \in B$. But we already know that $\overline{a} \notin B$. Hence, this is a contradiction.

In the second case we have $\overline{b} \neq a$. From this and from $\overline{b} \in C$ we know, by definition of sub-model, that $b \in \mathrm{sm}(C, a)$. Hence, $B \cap \mathrm{sm}(C, a)$ is not empty.

Hence, there is a model M for S such that $\mathrm{sm}(A, a) \subseteq M$. □

This lemma is more powerful then the Independent Blocked Clause Rule, because each blocked clause is nondecisive but not the other way around.

6. The Independent Strongly Nondecisive Clause Rule

In this section we introduce the Independent Strongly Nondecisive Clause Rule, a generalization of the Independent Nondecisive Clause Rule. This rule is introduced by the author.

The Independent Strongly Nondecisive Clause Rule states that if a clause set contains an independent strongly nondecisive clause, then it is satisfiable and a sub-model generated from this clause and from one of its strongly nondecisive literals is a partial model, i.e., we can simplify the clause set by propagating this sub-model. These requirements are fulfilled very often by 3-SAT benchmark problems, but checking independent-ness and strongly nondecisive-ness is expensive.

We will see from our test result that the Independent Blocked Clause Rule can be applied only on few 3-SAT instances. The Independent Nondecisive Rule is better, but still can be applied only on every tenth benchmark problem. Therefore, we tried to find an even more powerful simplification rule. Finally, we found the Independent Strongly Nondecisive Clause Rule.

The idea is the following: We know that a nondecisive clause is either blocked or a special construction ($\mathrm{Res}(A,B) \cup \{a\}$) is subsumed. This rings a bell. If we would use the notion of entailed instead of subsumed then the rule would be more powerful. Let us check whether this idea works or not.

Lemma 6.1 (Independent Strongly Nondecisive Clause Rule). *Let S be a clause set. Let $A \in S$ be strongly nondecisive in S and independent in $S \setminus \{A\}$. Let $a \in A$ be a strongly nondecisive literal in A, S. Then there is a model M for S such that $sm(A,a) \subseteq M$.*

Proof. We know that A is independent in $S \setminus \{A\}$. Hence, by definition of independent, we know that there is a clear clause C that is subsumed by A and not subsumed by any other clause in S. Since $A \subseteq C$ we know that $\mathrm{sm}(A,a) \subseteq \mathrm{sm}(C,a)$. Hence, it suffices to show that $\mathrm{sm}(C,a)$ is a model for S. To show this, by definition of model, it suffices to show that for an arbitrary but fixed $B \in S$ we have that $B \cap \mathrm{sm}(C,a)$ is not empty. There are three cases, either (a) $a \in B$ or (b) $\overline{a} \in B$ or (c) $a \notin B$ and $\overline{a} \notin B$.

In case (a) we have $a \in B$. From this and from the definition of sub-model we know that $a \in B \cap \mathrm{sm}(C,a)$.

In case (b) we have $\overline{a} \in B$. From this and from $a \in A$ is nondecisive in A, S, by definition of nondecisive literal, we know that either there is a literal $b \in B$ which has $b \neq \overline{a}$ and $\overline{b} \in A$ or $\mathrm{Res}(A,B) \cup \{a\})$ is entailed in $S \setminus \{A\}$.

In the first case we know, by definition of sub-model, that $b \in \mathrm{sm}(A,a)$.

In the second case we know that $\mathrm{Res}(A,B) \cup \{a\})$ is entailed in $S \setminus \{A\}$. From this we know, by definition of entailed, that

$$\forall [D \in CC][A \cup B \setminus \{\overline{a}\} \subseteq D] \exists [E \in S][E \neq A] E \subseteq D.$$

From this we know that there is a literal $b \in B, b \neq a$ such that $b \notin C$ because otherwise we would have that $A \cup B \setminus \{\overline{a}\} \subseteq C$, which would mean that C is subsumed in $S \setminus \{A\}$, which would be a contradiction. From $b \notin C$ we know, by definition of clear clause, that $\overline{b} \in C$. From $b \neq a$ we know, by definition of sub-model, that $b \in \mathrm{sm}(C,a)$. Hence, $b \in B \cap \mathrm{sm}(C,a)$.

In case (c) we have $a \notin B$ and $\overline{a} \notin B$. Since C is not subsumed by $S \setminus \{A\}$ we know, by definition of subsumption, that $B \not\subseteq C$. From this, since C is a clear clause we know that for some $b \in B$ we have $\overline{b} \in C$. There are two cases, either $\overline{b} = a$ or $\overline{b} \neq a$.

In the first case we have $\overline{b} = a$, i.e., $\overline{a} \in B$. But we already know that $\overline{a} \notin B$. Hence, this is a contradiction.

In the second case we have $\overline{b} \neq a$. From this and from $\overline{b} \in C$ we know, by definition of sub-model, that $b \in \mathrm{sm}(C,a)$. Hence, $B \cap \mathrm{sm}(C,a)$ is not empty.

Hence, there is a model M for S such that $\mathrm{sm}(A,a) \subseteq M$. □

Note that $\mathrm{Res}(A,B) \cup \{a\} = A \cup B \setminus \{\overline{a}\}$.

We see that this proof is almost the same as the proof of the Independent Nondecisive Clause Rule except for the second part of case (b). Here we use the

following idea: C is subsumed by A but not by $A \cup B \setminus \{\overline{a}\}$, hence there is a literal $b \in B$ which has $b \neq a$ and $b \notin C$.

So the Independent Strongly Nondecisive Clause Rule works. But to decide whether we can apply it or not we have to perform an entailed-ness check, which is an exponential time method.

What can we do? There are some special cases when it is easy to check entailed-ness. For example the clause E is entailed in the clause set S if we have $E \in S$ or there is a clause $B \in S$ which simply subsumes E. This cases are very rare. The case we are going to describe occurs very often in 3-SAT problem instances.

Assume that we want to check whether the clause E is entailed in the clause set S. Assume we found a clause $D \in S$ which has the following two properties: (a) $\mathrm{diff}(E, D) = \emptyset$ and (b) $D \setminus E$ is a singleton. The first property is needed otherwise D could not subsume any clear clause subsumed by E. The second property says that D subsumes the "half" of E.

Assume that $D \setminus E = \{d\}$. Then D subsumes all clear clauses which are the superset of $E \cup \{d\}$. If E subsumes $2k$ clear clauses and $d \notin E$ then $E \cup \{d\}$ subsumes k clear clauses and $E \cup \{\overline{d}\}$ subsumes the remaining k clear clauses. Hence, we can say that D subsumes the "half" of E. So we can reduce the problem to whether $E \cup \{\overline{d}\}$, the remaining "half", is entailed in S or not. We call this step to cut E in half.

This situation occurs very often in 3-SAT problem instances, because our $E = A \cup B \setminus \{\overline{a}\}$ has a length of 5, clauses in the input clause set have a length of 3, and usually we have $n \gg 5$, where n is the number of variables. This means that it is very likely that we can use this step at least once.

The following algorithm uses this step to find strongly nondecisive clauses. In the worst-case it is a $O(n^3m^2)$ time method, but there is no guarantee that it finds any strongly nondecisive clauses.

GetSNDClauses

```
1  function GetSNDClauses(S : clauseset) : array of ⟨clause, literal⟩
2  begin
3    i := 0;
4    // We need i to index the array SND.
5    for each A ∈ S do
6      a_is_snd := False;
7      for each a ∈ A while a_is_snd is False do
8        B_snds_a := True;
9        for each B ∈ S, ā ∈ B while B_snds_a is True do
10         b_blocks_a := False;
11         D_subsumes_E := False;
12         B := B \ {ā};
```

```
13          if (diff(B, A) ≠ ∅) then b_blocks_a := True;
14          else
15            E := A ∪ B;
16            for each D ∈ S, D ≠ A while D_subsumes_E is False do
17              if (D ⊆ E) then D_subsumes_E := True;
18              if (diff(D, E) = ∅ ∧ |D \ E| = 1) then
19                E := E ∪ (D \ E);
20                Restart the last loop ;
21                // We have to restart the loop on clauses D,
22                // because the remaining half could be subsumed
23                // by a clause, which was already considered.
24              fi
25            od
26          fi
27          if (¬b_blocks_a ∧ ¬D_subsumes_E) then B_snds_a := False;
28        od
29        if (B_snds_a) then a_is_snd := True;
30      od
31      if (a_is_nond) then (SND[i], i) := (⟨A, a⟩, i + 1);
32    od
33    return SND;
34  end
```

The new rows are the ones from 14 till 26. We use in the 20th row a very interesting solution, we restart the innermost loop. We discuss this issue a bit later.

One can see that this algorithm returns an array of ordered pairs. An ordered pair contains a strongly nondecisive clause C and a strongly nondecisive literal $c \in C$.

Note that this algorithm might not find all strongly nondecisive clauses, because it does not use entailed-ness check, but the "cut E in half" step, described above.

This algorithm is an $O(n^3m^2)$ time method in the worst-case, where n is the number of variables and m is the number of clauses of the input clause set. It is an $O(n^3m^2)$ time method, because we have two loops on clauses and two loops on literals, but the innermost loop might be restarted n times in the worst-case.

One might ask, why do we need to restart the innermost loop? Assume we have the situation that we can cut E in half, i.e., we have found a clause $D \in S, D \neq A$ which has $\text{diff}(D, E) = \emptyset$ and $D \setminus E$ is a singleton. Then there is no D' clause among the ones we already considered such that D' subsumes $E \cup (D \setminus E)$, because D' fulfills the same requirements as D, i.e., it would be already used to cut E in

half. Then why should we restart?

That is true, but there might be clauses among the ones we already considered which can cut the new E in half and in the rest of the clause set there is no suitable clause which subsumes E or can cut it in half. Therefore, we have to restart the innermost loop.

7. Heuristics

In this subsection we introduce three heuristics. All of them are suitable more or less to guess whether a clause is independent or not.

All three heuristics are based on the following idea. A clause A is independent in the clause set $S \setminus \{A\}$ if $\overline{A}$ is a subset of a model of S, i.e., after propagating $\overline{A}$ on S, let us call the resulting clause set S', S' is satisfiable. Of course we do not want to perform expensive satisfiability checks, but we want to guess whether it is satisfiable or not. The idea is the following: the less clauses are contained in S', the more likely is that it is satisfiable.

This means that we have to count the clauses in S'. But propagation of an assignment is still to expensive for us. Therefore, we count the clauses in the following set:

$$\{B \mid B \in S \wedge \mathrm{diff}(A, B) = \emptyset\}.$$

Note that if a clause C is in this set then the clause $C' = C \setminus A$ is element of S'.

In the first version, called $IBCR$-1111, we just count each blocked clause A the clauses B that have $\mathrm{diff}(A, B) = \emptyset$ and we choose the one for which this number is the smallest.

Our test results on 3-SAT problem instances shows that this heuristic provides an independent blocked clause in 68% of the cases if there is an independent blocked clause.

In the other two versions we use weights.

In the second version, called $IBCR$-1234, we count each blocked clause A the clauses B which has $\mathrm{diff}(A, B) = \emptyset$ and we choose the one for which this number is the smallest. But we count clauses B with different weights. The weight W_B is

$$W_B := 1 + |A \cap B|.$$

For example if A is a 3-clause and $|A \cap B| = 2$ then $W_B = 3$.

Our test results on 3-SAT problem instances shows that this heuristic provides an independent blocked clause in 71% of the cases if there is an independent blocked clause.

In the third version, called $IBCR$-1248 the weight W_B is

$$W_B := 2^{|A \cap B|}.$$

For example if A is a 3-clause and $|A \cap B| = 2$ then $W_B = 4$.

Our test results on 3-SAT problem instances shows that this heuristic provides an independent blocked clause in 73% of the cases if there is an independent blocked clause.

After this short overview we give more details. First we have to explain the names of the three heuristics: $IBCR$-1111, $IBCR$-1234, and $IBCR$-1248. The word "IBCR" is just the abbreviation of Independent Blocked Clause Rule.

We have tested these heuristics on 3-SAT problem instances, where $|A \cap B|$ can be 0, 1, 2, or 3. The remaining part of the names comes from the values of weights. In the first heuristic the weight is the constant 1. Therefore, its name is $IBCR$-1111. In the second one the weight is defined by $1+|A \cap B|$, i.e., the weights are 1, 2, 3 or 4, respectively. Therefore, its name is $IBCR$-1234. In the third one the weights are 1, 2, 4, 8, respectively. Therefore, its name is $IBCR$-1248.

We present the pseudo-code of the third variant. This algorithm is an $O(n^2m^2)$ time method in the worst-case, where n is the number of variables and m is the number of clauses in the input clause set. It is an $O(n^2m^2)$ time method, because we have two loops on clauses and other two on literals.

IBCR-1248

```
1  function IBCR-1248(S : clause set) : ⟨clause, literal⟩
2  begin
3    min_Counter := Infinite;
4      // The variable min_Counter stores the minimum value of Counter.
5      // First time should be big enough.
6    for each A ∈ S do
7      a_is_blocked := False;
8      for each a ∈ A while a_is_blocked is False do
9          // Here begins the code which is relevant for the heuristic
10       Counter := 0;
11       B_blockes_a := True;
12       for each B ∈ S while B_blocks_a is True do
13         if (diff(A, B) = ∅) then Counter := Counter + 1 * (2^|A∩B|);
14           // The weight is 2^|A∩B|.
15         if (ā ∉ B) then continue ;
16           // Remember, we have to visit all B ∈ S which has ā ∈ B
17           // to decide whether a ∈ A is blocked or not.
18         b_blocks_a := False;
19         for each b ∈ B, b ≠ ā while b_blocks_a is False do
20           if (b̄ ∈ A) then b_blocks_a := True;
21         od
22         if (¬b_blocks_a) then B_blocks_a := False;
```

```
23          od
24          if (B_blocks_a) then a_is_blocked := True;
25          if (a_is_blocked and (Counter < min_Counter)) then
26            (min_Counter, min_A, min_a) := (Counter, A, a);
27          fi
28        od
29      od
30      return ⟨min_A, min_a⟩;
31  end
```

From this algorithm one can easily construct the other two or even other heuristics.

We can see that this heuristic returns a clause, say C, and a literal, say c. The clause C is a blocked clause and the literal c is a blocked literal in it. The heuristic state that C is independent. But this might be false.

If it is true, then it is fine because we can simplify our input clause set by a sub-model propagation using $\mathrm{sm}(C, c)$.

If it is false, then we still can gain something. We can add a shorter clause than C, because, by the Lucky Failing Property of Sub-Models, we know that $C \setminus \{c\}$ is entailed by the input clause set.

We do not know which case will be applied but we hope that the first one occurs more frequently.

These heuristics do not use the fact that the clause is blocked or not. Therefore, we can generalize them very easily for guessing independent-ness of (strongly) nondecisive clauses.

In the names of these heuristics we use the following acronyms: INCR for Independent Nondecisive Clause Rule; ISNCR for Independent Strongly Nondecisive Clause Rule.

8. Test results

In this section we describe shortly our java implementation of the simplification rules and we present the test results we have got on problems from the SATLIB problem library.

Our java implementation has three classes, Clause, ClauseSet and Satisfiable. The class Clause contains two BitSet objects, *positive* and *negative*. If we represent a clause where the first variable occurs positively then the first bit of the BitSet *positive* is set (1) and the first bit of BitSet *negative* is clear (0). This means that our implementation is close to the Literal Matrix View.

This implementation is not competitive with the newest SAT solvers because it does not use enhanced data structures or techniques like back jumping but it is

good enough to test whether the simplification rules can be applied on benchmark problems or not.

We have tested the heuristics on Uniform Random-3-SAT problems [6] from the SATLIB – Benchmark Problems homepage:
`http://www.intellektik.informatik.tu-darmstadt.de/SATLIB/benchm.html`

We used the smallest problem set, uf20-91.tar.gz, which contains 1000 problems, each has 91 clauses and 20 variables and is satisfiable.

We used a Pentium 4, 2400 MHz PC machine with 1024 MB memory to perform the tests.

Here we present our test results for the problems of uf20-91.tar.gz as a table ($IBCR$: Independent Blocked Clause Rule, $INCR$: Independent Nondecisive Clause Rule, $ISNCR$: Independent Strongly Nondecisive Clause Rule):

	$IBCR$	$INCR$	$ISNCR$	from
SND clauses:	601	1128	61122	91000
Problems with SND:	256	465	1000	1000
Independent SND:	77	125	4011	91000
Prob.s with indep. SND:	60	102	951	1000
X-1111:	41 / 60	61 / 102	89 / 951	
X-1234:	43 / 60	72 / 102	142 / 951	
X-1248:	44 / 60	76 / 102	166 / 951	

By "SND clauses" we mean in the column of Independent Blocked Clause Rule blocked clauses, in the next column nondecisive clauses, and in the next column strongly nondecisive clauses. The column "from" shows how many clauses and clause sets, respectively, do we have in total.

The line X-1111: 41 / 60 61 / 102 89 / 951 means that: $IBCR$-1111 successfully guesses 41 times an independent blocked clause from the 60 cases where we checked whether we have independent blocked clauses; $INCR$-1111 is successful 61 times from 102; and $ISNCR$-1111 is successful 89 times from 951.

Now we give the same table but the results are given in percentages.

	$IBCR$	$INCR$	$ISNCR$	from
SND clauses:	0.66%	1.23%	67.16%	91000
Problems with SND:	25.6%	46.5%	100%	1000
Independent SND:	0.08%	0.13%	4.4%	91000
Prob.s with indep. SND:	6%	10.2%	95.1%	1000
X-1111:	68.334%	59.8%	9.35%	
X-1234:	71.667%	70.58%	14.93%	
X-1248:	73.334%	74.5%	17.45%	

We can see that the X-1248 is the best heuristic, but still it could guess an independent strongly nondecisive clause only in 17% of the cases where we know that there are some.

It is so because it is very hard to guess independent clauses. We have better results in the other two cases because there are a lot of instances where we have only one or two independent blocked or nondecisive clauses. One can see that only the 0.66% of clauses are blocked while 67% are strongly nondecisive.

We believe that these simplifications are very useful, because if it turns out that the selected blocked clause is not independent, after propagating a sub-model generated from it, then we can still, by the Lucky Failing Property of Sub-Models, add a shorter clause to our clause set.

References

[1] Aspvall, B., Plass, M.F., Tarjan, R.E., A linear-time algorithm for testing the truth of certain quantified boolean formulas, *Information Processing Letters*, 8(3) (1979) 121–132.

[2] Aspvall, B., Recognizing disguised NR(1) instances of the satisfiability problem, *J. of Algorithms*, 1 (1980) 97–103.

[3] Boros, E., Hammer, P.L., Sun, X., Recognition of q-Horn formulae in linear time, *Discrete Applied Mathematics*, 55 (1994) 1–13.

[4] Boros, E., Crama, Y., Hammer, P.L., Saks, M., A complexity index for satisfiability problems, *SIAM J. on Computing*, 23 (1994) 45–49.

[5] Chandru, V., Hooker, J., Extended Horn sets in propositional logic, *J. of the ACM*, 38(1) (1991) 205–221.

[6] Cheeseman, P., Kanefsky, B., Taylor, W.M., Where the really hard problems are, *Proceedings of the IJCAI-91*, (1991) 331–337.

[7] Cook, S.A., The complexity of theorem-proving procedures, *Proceedings of the 3rd ACM Symposium on Theory of Computing*, (1971) 151–158.

[8] Dalal, M., Etherington, D.W., A hierarchy of tractable satisfiability problems, *Information Processing Letters*, 44 (1992) 173–180.

[9] Dowling, W.F., Gallier, J.H., Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. of Logic Programming*, 1(3) (1984) 267–284.

[10] Even, S., Itai, A., Shamir, A., On the complexity of timetable and multicommodity flow problems, *SIAM J. on Computing*, 5(4) (1976) 691–703.

[11] Gelder, A.V., Propositional search with k-clause introduction can be polynomially simulated by resolution, *Proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, 1998.

[12] Knuth, D.E., Nested satisfiability, *Acta Informatica*, 28 (1990) 1–6.

[13] Kullmann, O., New methods for 3-SAT decision and worst-case analysis, *Theoretical Computer Science*, 223(1-2) (1999) 1–72.

[14] Kullmann, O., On a generalization of extended resolution, *Discrete Applied Mathematics*, 96-97(1-3) (1999) 149–176.

[15] Kusper, G., Solving the SAT problem by hyper-unit propagation, *RISC Technical Report 02-02*, University Linz, Austria, (2002) 1–18.

[16] Kusper, G., Solving the resolution-free SAT problem by hyper-unit propagation in linear time. *Annals of Mathematics and Artificial Intelligence*, 43(1-4) (2005) 129–136.

[17] Kusper, G., Finding models for blocked 3-SAT problems in linear time by systematical refinement of a sub-model, *Lecture Notes in Artificial Intelligence 4314, KI 2006: Advances in Artificial Intelligence*, (2007) 128–142.

[18] Schlipf, J.S., Annexstein, F., Franco, J., Swaminathan, R.P., On finding solutions for extended Horn formulas, *Information Processing Letters*, 54 (1995) 133–137.

[19] Scutella, M.G., A note on Dowling and Gallier's top-down algorithm for propositional Horn satisfiability. *J. of Logic Programming*, 8(3) (1990) 265–273.

[20] Tovey, C.A., A simplified NP-complete satisfiability problem, *Discrete Applied Mathematics*, 8 (1984) 85–89.

Gábor Kusper
Lajos Csőke
Gergely Kovásznai
Institute of Mathematics and Informatics
Eszterházy Károly College
P.O. Box 43
H-3301 Eger
Hungary

e-mail:
gkusper@aries.ektf.hu
csoke@aries.ektf.hu
kovasz@aries.ektf.hu

Annales Mathematicae et Informaticae
35 *(2008) pp. 95–121*
http://www.ektf.hu/ami

Decision based examination of object-oritented methodology using JML

Szabolcs Márien

University of Debrecen, Debrecen, Hungary

Submitted 28 September 2007; Accepted 10 October 2008

Abstract

The aim of object-oriented conception is to make sure that the program is well-structured, so as to become perspicuous; it can be extended easily, so that it could be maintained more easily; and its reusability can be increased in order to be modularized. There are lots of measuring methods, by which the realization of the mentioned aims is measured. The measuring methods are the metrics that give us indicators showing the complexity of the program structure.

Can the existing object-oriented metrics really indicate the structural quality of the program? As we know, these metrics examine structural properties like the number of inheritance levels, the number of subclasses, or the number of methods, which can not be the basis of real quality examinations. The reason of this is that the aim of the object-oriented conception is not clarified. In order to realize the aims of object-oriented technology, object-oriented paradigms should be reinterpreted.

According to our new conception object-oriented methodology is based on the elimination of decision repetitions, that is, sorting the decisions to class hierarchy, so that the data structure and methodology of the decision options could be determined by the subclasses of the given class. When sorting the decisions and decision options to a class and its subclasses, only the first decision case will be executed, which will be archived and enclosed by the instantiation of one of the subclasses. For the following decision cases the archived decision result can be used without knowing which decision option was used, that is, which subclass was instantiated, as it is enclosed by using the type of the parent class, except the necessary data structure and/or methodology is decision option specific.

There are two states of decisions depending on the place of their defining: the decision options and their data structures and methodologies can be defined in the method, but the sorted decision can be defined by a class and its subclasses.

In order to support the practical benefit of our conception, we are going to show how decisions can be formalized (that is, whether the decision states are defined in a method or by a class hierarchy) based on JML. Using the JML formalization those cases should be identified where decisions can be sorted, thus the elimination of decision redundancy is suggested.

According to our new conception the aim of object-oriented technology is the elimination of decision repetitions, which can be realized by sorting decisions. Therefore inheritances are the abstract definition forms of decisions, so the inheritances can be interpreted as decision abstractions.

1. Motivation

Object-oriented programming is a programming methodology. The programs based on it organize the collaborations of objects that are instances of one of the classes. Classes are built in class-hierarchies, where the connections between the classes are realized by inheritance relationships. [1]

The base paradigms of object-oriented technology are encapsulation [3, 6], inheritance [3, 4, 6], polymorphism [3, 6] and message-passing [3, 4, 6]. Encapsulation means that the data structures and the methodology are defined together, enclosing them in units as objects. Encapsulated data structures and methodology can be defined in classes, the instances of which are called objects. Modularized construction can be realized with the help of encapsulation, and as a result – if the methodology of one of the objects is changed – there are no side-effects in other objects. [6] Inheritance means that the data structures and the methodology, defined in a class, can be inherited by its subclasses. Subclasses can define new data structures and methods as complements of the inherited properties [4, 5] and can override inherited data structures and methodologies. Polymorphism means that the classes' methods can be overridden by their subclasses, so the method, which gets the control, is selected just in runtime (Late Binding). [3] Late Binding – according to another terminology – means that an object sends similar messages to different objects (classes and their subclasses), which results in the execution of a different code. [6]

There are lots of metrics in order to control the programs' quality. The quality of the design, the program and the efficiency of the testing can be checked by using these metrics. [4] The metrics defined in [2] (MOOD) and [4] are based on object-oriented paradigms. Accordingly, these metrics are used as a base concept of encapsulation (MHF[2], AHF[2]), inheritance (MIF[2], AIF[2], DIT[4], NOC[4]), polymorphism (POF[2]), message-passing (COF[2], LCOM[4], CBO[4]) in order to check the software quality.

But the metrics based on the base paradigms can be used for checking the quality subsequently. If the result of the checking shows that structure of the program is bad, it can be repaired by reconstruction. Using the solution in [7] reconstruction can be solved automatically based on inheritance checking.

By supporting the work of the program designer in the designing phase lots of designing failures could be avoided, and designing experience as designing 'recipes'

could be reused. According to this concept, the creation of quality programs is guaranteed by rules based on designing experience. These generic recipes are Design Patterns [8], which is used to create a quality design and program. Are there any appropriate answers for the aims of Design Patterns beyond the general descriptions? We do not think there are, as the profession tried to define the rules of the programs' quality by collecting Design Patterns, but there are no clear answers for what the main concepts of Design Patterns are.

What is the reason for the deficiency of object-oriented metrics and why are not there any clear answers for the aims of Design Patterns? The answers can be found in [11]. According to that conception, the grounds for the answers can be found in the existing interpretation of object-oriented paradigms that block the extended examination of object-oriented methodology. Between the metrics – which are based on the interpretations of these paradigms – and the program quality there is no obvious connection, because these metrics depict the complexity of the program. Nevertheless, there are no clear answers for the aims of Design Patterns based on the existing interpretations of paradigms.

In order to resolve the problems described in [11], a new interpretation of the basic object-oriented paradigms is described, by which the basic concepts of object-oriented methodology can get another approach. According to this, we give new options for controlling the program quality and for repairing the programs as new guidelines are realized (Introduction, [11]) that improve the structures of programs and make their maintenance.

The new conception gives Design Patterns a clear interpretation. Accordingly, Design Patterns give us recipes for accomplishing the requirements of well-structured programs by reducing the number of decision repetitions. So Design Patterns give us recipes how decision repetitions can be eliminated in different decision construction cases [11].

In order to examine program structures and the performance of the guidelines of well-structured programs, we need a formalization tool that examines the definitions of decisions. Formal examinations are based on the Java programs' behaviour interface specification language – JML [12, 13, 14, 15]. JML specifies the data structures and the methodology of decision options based on logical expressions. JML formalized decisions have already been examined according to the decision-based conception and the guidelines of well-structured programs.

2. Introduction

Based on the decision-based interpretation of object-oriented concept [11], in this paper a new formalization method of decisions is realized using JML.

In this section, according to [11], the new interpretation of object-oriented concept is shown.

The decisions of the program code decide about the data structure and functionality are specified in the decisions. The main concept of object-oriented methodology is the elimination of decisions' repetition by sorting them to a "common

place". This "common place" is a class with subclasses, so decision repetition can be eliminated by class hierarchy. After sorting the decisions, the decision about the necessary functionality and data structure is executed only once. Decision archiving is realized by the instantiation of the subclass with the appropriate functionality and/or data structure. The result of the decision (the archived decision) - as an instance of the appropriate subclasses - can be used at other decision places without having any specific information about it. Accordingly, the decisions can be enclosed in class hierarchy.

Decision cases are important parts of programs, where the appropriate decision option can be decided by using the actual values.

In order to ensure that a program is well-structured, we should note the following:

- The methodology and/or the data structure of the decision options have to be defined just once, so the code of the decision options will be defined just once, unless sorting the decisions is impossible. It is important to consider manageability, because the introduction of a new decision option can be solved easily if it can only be built in one place of the program.

- Decisions having equivalent decision predicates but differing in their decision option definitions should not reoccur. This case is different from the previous one, as though the decision predicates are equivalent, the methodology and/or the data structures of the decision options are different. In these cases the decisions can be contracted too, so the definitions of the different decision options can be defined by contracting them in the same class hierarchy according to the decision predicates.

- Decisions should not reoccur, so a decision should be executed just once during the same running, if the predicates of decisions are equivalent and the decision options define the same data structure and functionality. The elimination of decision repetitions has two aspects:

  - The result of the decision - as the data structures or/and the methodology of the decision options - can be used several times.

  - The result of the decision can be used later more times, but a new instance of the structures or/and methodology of the decision options is created each time. The archived decision can be used later for creating an instance of the decision options.

In order that the analysis could be realized based on the decisions, it is important how the basic paradigms of object-oriented technology (inheritance, polymorphism, encapsulation) and its basic tools (class hierarchy, aggregation) can be joined to the decision based concept.

2.1. Inheritance as decision abstraction

Inheritance means that the data structure and the methodology defined in a class can be inherited by its subclasses. The subclasses can define new data structures and methods as complements of the inherited properties [4, 5] and can overwrite the inherited data structure and methodologies.

The decision can choose the running program code and the data structure. In order that a decision could be archived, it has to be sorted, which means that the data structure and methodology of the decision options have to be defined in a class hierarchy, as a parent class and its subclasses. Derivation/inheritance ensures the enclosing and archiving of the decision to the class hierarchy, therefore the definitions of the decisions can be contracted and decision repetitions can be eliminated.

According to this interpretation class hierarchy – the class with its subclasses – based on inheritance is the abstract form of the decision.

If the decision is defined in a class hierarchy, the following is realized:

- Elimination of the code repetition, which defines the decision options, so the conditions of the decision options can be defined just one time.
- Archiving the decision, so that the result of the decision could be used for the next occasions, unless the required data structure or methodology is specified by one of the decision options only.
- Enclosing the decision. The result of the decision is not known in the next decision cases, unless the required data structure or methodology is specified by only one of the decision options.
- By introducing a new subclass, decision options can be extended easily. When creating a new subclass, only the first decision case has to be fit for handling the new decision option, because the decision will be enclosed on the next occasions, unless the required data structure or methodology is specified by only one of the decision options.

As it can be seen, if the data structure or/and the methodology is specified by just one of the decision options, the advantages of decision sorting can only be realized partially. The forceful usage of polymorphism can completely realize the advantages of decision sorting from the point of view of inheritance.

2.2. Polymorphism as decision enclosing

If the decision is realized in the first decision case, one of the subclasses will be instantiated based on the chosen decision option. The instance of the appropriate subclass archives the decision and the visible type of the instance will be the parent class of the subclass. With this, the enclosing of the decision can be realized, because the result of the decision can be used without of the knowledge of the decision on the next occasions.

2.3. Encapsulation

Decision options can be defined by data structure and methodology. The decision is defined in a method, if the appropriate If-Else command's blocks define the data structure and the methodology of the decision options. If the decision is defined in an abstract form sorted in class hierarchy, the decision options are realized in the subclasses. If there is a change in the data structure and the methodology of the decision option, no side-effects occur in other decision cases and other decision options, accordingly the decision option can define the data structure and the methodology by a subclass enclosing them (the data structure and the methodology).

2.4. Aggregation as dynamic decision embedding

By aggregation the sorted decision can be referred to. If there is a decision case, in which the appropriate decision option is chosen (with the proper data structure and methodology), and next time the operations are executed based on the chosen methodology and data structure as the result of the decision, the sorted decision can be used in the decision cases. The result of the decision will be referred to by aggregation.

When we talk about aggregation, we have to know that it is the tool of relating decisions.

In the following sections of the paper we will show how the described decision based conception can be supported by JML. In Section 3 the JML specification language is described, and on the basis of this, Section 4 introduces the formalization method of the two states of the decisions (defined by method or by class hierarchy). In the final part of the paper, in Section 6, an example shows how the decisions can be formalized before and after decision sorting.

3. JML

JML – Java Modelling Language is a behaviour interface specification language [12, 13], by which the syntactical interface and the behaviour of Java programs is specified. [12]

The syntactical interfaces are the Java interfaces and the programmer interfaces of Java programs, that is, the signatures of the methods, the names and types of the variables. The behaviour of the interfaces can be specified by JML annotations, which define how classes and methods can be used. [12]

The JML specification language combines the Eiffel-style syntax with the model-based semantics as in VDM and Larch. Eiffel-style assertions are extended to use Java expressions. JML combines this with the model-based approach of VDM and Larch. [13, 15]

Accordingly, JML contains many state-based specification languages' core specification constructions, for example, pre- and post-conditions, assertions, invariants. These constructions are not able to realize the formal modular verification of object-oriented programs. Therefore, JML uses extra constructions such as frame-properties, data groups, ghost and model variables. [14]

JML specifications can either be written in separate – specification – files or as annotations in Java program files (the Java compiler interprets these annotations as comments, which are ignored by the compiler). Specification files and their specifications can be organized into inheritance-hierarchies, which make the creation of the well-structured specification easier.

There are two kinds of specification cases in JML: Lightweight and Heavyweight specifications. Lightweight specification cases are useful when giving partial specifications, but if the complete specification is necessary, we should use the heavyweight specification option.

In the following part the main concepts of JML specification constructions are described, in order that the Reader could interpret the examination of the decision-based extension of object-oriented concepts by JML more easily.

There are two kinds of specification constructions of JML:

- Behaviour specification constructions, such as 'assert', 'assume', 'require', ...
- Specification constructions of classes and interfaces, such as invariants, models, ...

3.1. Behaviour specification constructions

The basic constructions of JML are the pre- and post-conditions of the commands and methods, which determine the program states before and after the executions of the commands or methods.

The pre- and post-conditions can be described as a contraction between a method (its implementer) and its caller (user) as follows:

- Pre-condition:
  - The method or the command assumes that the pre-condition has been realized.
  - The caller of the method or the command ensures the realization of the pre-condition.
- Post-condition:
  - The method or the command ensures the realization of the post-condition.
  - The caller of the method or the command assumes the realization of the post-condition.

The appearances of the pre- and post-conditions in JML specification are:

- Conditions in the methods:
  - 'Assume': Assertion that the program requires.
  - 'Assert': Assertion that the program ensures.
- Conditions between methods:
  - 'Requires': It specifies the pre-condition of the method.
  - 'Ensures': It specifies the post-condition of the method.

The pre- and post-state of the variables can be distinguished as follows:

- Pre-state: The starting state of the variables is signed by enclosing the variable name with the '\old()' expression.
- Post-state: The ending state of the variables is signed by the variable name.

The variables with modified values in the methods are specified by assignable annotations such as frame conditions, which can define the "frame" of the possible state-transitions. JML behaviour specification constructions are based on the requirements of Hoore calculus.

3.2. Specification of interfaces and classes

JML can specify invariants, such as general conditions, that help to narrow the state-space of classes and interfaces.

The initial conditions of the variables can be specified.

History constraint specifies the relations between pre- and post-states, which are realized by every state-transition.

The data group is a set of fields (locations). The data-groups, such as the grouped fields, are the basic-units of the states and the state-transitions.

JML has an abstract construction. It is the model variables that can be used in the model specification. The 'represents' clause can join the model variable with the implementation variable as its implementation representation.

4. Decision formalization

In order to formalize the decisions of object-oriented programs, the formalization of the data structure and the behaviour of the programs can be solved, because it is necessary to compare the data structure and the behaviour of decision options. The analysis of decision predicates [11] is necessary for the examination of redundant decisions.

JML has constructions to realize behaviour specification and the specification of data structures. Because the behaviours are specified by logical formulae as post-conditions, the equivalence of the decisions' decision options can be examined based

on the data structures and the behaviours. Based on the formalized behaviours as post-conditions, the examination of the decision-predicated are realized, too.

In this section the formalization of the decisions by JML is described. We describe the JML formalization of one-level decisions and the formalization of two- or more-level decisions.

In order to show the connections between the two states of the decisions (the decision is defined by a method or by class hierarchy), we describe the JML formalization of non-sorted and sorted decisions. Based on the formalized sorted decisions, we can see how the result of the first decision case is in-closed, archived, which can be reused further on, in other decision cases of that decision.

4.1. Formalization of non-sorted decisions

If the decision is not sorted, the methodology and the data structure of the decision options are defined in the method, not in the class hierarchy.

The pre-conditions and the post-conditions of the behaviours and the consequences of the decision options are defined in the specification of the method, where the definition of the decision options is separated by the keyword 'also'.

```
/ * @ public normal_behavior //D
   @ requires p1; //DL1
   @ assignable v1,..., va, vb, ..., vc;
   @ ensures condition1 & & ... & & conditioni & & conditionj & & ... & & conditionk;
   @ also
   @ requires !(p1); //DL2
   @ assignable v1,..., va, ve, ..., vf;
   @ ensures condition1 & & ... & & conditioni & & condition1 & & ... & & conditionm;
@ * /
```

The two decision options of the decisions are separated by the keyword 'also'.

The pre-conditions of the decision options are $p_1, !p_1$. The pre-condition determines the appropriate decision option, by which the appropriate data structure and behaviour is realized. The 'assignable' assertion defines the data structure, which is modified by the decision option. The behaviour of the decision option is defined by 'ensures' assertions as post-conditions.

The data structures and the behaviour of the decision options have common and decision option-specific parts. It is important, because if we sort the decisions, the common parts of the decision options are specified by the parent class in the class hierarchy, and the decision option specific parts are defined by the subclasses. The data structures of decision options are:

$v_1, \ldots, v_a$ – Variables, which are modified by all decision options.

$v_b, \ldots, v_c$ – Variables, which are modified in D_{L_1} decision option.

$v_e, \ldots, v_f$ – Variables, which are modified in D_{L_2} decision option.

The behaviours of the decision options are:

condition_1 &&$\ldots$&& condition_i – Common behaviours of the decision options.

condition_j && ... && condition_k – Behaviour, which is specified by D_{L_1} decision option.

condition_1 && ... && condition_m – Behaviour, which is specified by D_{L_2} decision option.

Formalization of the decision which contains other decisions (Complex decision).

```
/*@ public normal_behavior //D2 in D1
  @ requires p1; //D1L1
  @ {|
  @   requires p2; //D2L1
  @   assignable v1,...,va,vb,...,vc, //D1L1
  @                 vg,...,vh,vi,...,vj; //D2L1
  @   ensures condition1 & & ... & & conditiono & & conditionp & & ... & & conditionq & & //D1L1
  @                 conditiont & & ... & & conditionu & & conditionv & & ... & & conditionw; //D2L1
  @ also
  @   requires !(p2); //D2L2
  @   assignable v1,...,va,vb,...,vc, //D1L1
  @                 vg,...,vh,vk,...,vl; //D2L2
  @   ensures condition1 & & ... & & conditiono & & conditionp & & ... & & conditionq & & //D1L1
  @                 conditiont & & ... & & conditionu & & conditionx & & ... & & conditiony; //D2L2
  @ |}
  @ also
  @ requires !(p1); //D1L2
  @ {|
  @   requires p2; //D2L1
  @   assignable v1,...,va,ve,...,vf; //D1L2
  @                 vg,...,vh,vi,...,vj; //D2L1
  @   ensures condition1 & & ... & & conditiono & & conditionr & & ... & & conditions & & //D1L2
  @                 conditiont & & ... & & conditionu & & conditionv & & ... & & conditionw; //D2L1
  @ also
  @   requires !(p2); //D2L2
  @   assignable v1,...,va,ve,...,vf; //D1L2
  @                 vg,...,vh,vk,...,vl; //D2L2
  @   ensures condition1 & & ... & & conditiono & & conditionr & & ... & & conditions & & //D1L2
  @                 conditiont & & ... & & conditionu & & conditionx & & ... & & conditiony; //D2L2
  @ |}
@*/
```

The decision options ($D_{1_{L_1}}, D_{1_{L_2}}$) of D_1 decision contain the decision options ($D_{2_{L_1}}, D_{2_{L_2}}$) of D_2 decision. Complex decisions can be specified just like simple decisions. There are common parts and decision option specific parts of the decision options' behaviours and data structures. The common and the decision option specific variables and conditions of behaviours – as it can be seen in the specification of simple decisions – are signed by indexes.

4.2. Formalization of sorted decisions

If the decision is sorted, the decision is specified by the parent class and its subclasses. The parent class defines the common parts of the decision options, and the decision option specific parts are defined by the subclasses. The parent class as a type can archive the decision result of the decision case, accordingly, the variable that encloses the decision gets the parent class type (in this case its type is 'o'). The further decision cases can use the o-variable – which encloses the decision – in order to achieve the functions of the decisions and decision options.

4.2.1. One-level sorted decisions

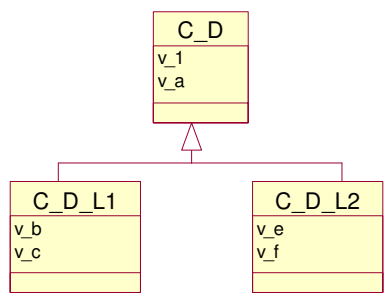

Diagram 1. C_D, $C_{D_{L_1}}$, $C_{D_{L_2}}$ class-hierarchy by UML diagram [9].

The JML formalization of sorted decision on the place of decision-sorting:

```
/*@ public normal_behavior //D
  @ requires o instanceof C_{D_{L_1}}; // ⇔ p_1 //D_{L_1}
  @ assignable o;
  @ ensures condition_1 & & ... & & condition_i & & condition_j & & ... & & condition_k;
  @ also
  @ requires o instanceof C_{D_{L_2}}; // ⇔!p_1 //D_{L_2}
  @ assignable o;
  @ ensures condition_1 & & ... & & condition_i & & condition_l & & ... & & condition_m;
@*/
```

There are not many differences between the formalizations of the decisions' two states (the sorted and the non-sorted states), because the formalization of the sorted decision shows the specification of decision options, too. So the decision

options of the "enclosed" decision are specified in the JML formulae of the decision. It is useful, because it shows the decision options of enclosed decisions.

In the first decision case, the decision predicate stays in its original form, but in the other decision cases – according to the decision based conception – the type of the object determines the behaviour of the decision (such as the behaviour of the appropriate decision option).

By sorting the decision, the behaviours of the decision options are separated in two subclasses, and the common parts are sorted into the parent class. The common and the decision option specific parts are united providing the appropriate behaviour in the decision cases.

The JML formalization of the parent class of the sorted decision is as follows:

```
/*@ public normal_behavior //D
  @  assignable v_1, v_a;
  @  ensures condition_1 && ... && condition_i;
@*/
```

The parent class provides only the common behaviour of the sorted decision.

The subclasses specify the decision option specific parts of the sorted decision completing the common behaviour:

```
/*@ also //D_L1
  @ public normal_behavior
  @  assignable v_b, v_c;
  @  ensures condition_j && ... && condition_k;
@*/

/*@ also //D_L2
  @ public normal_behavior
  @  assignable v_e, v_f;
  @  ensures condition_l && ... && condition_m;
@*/
```

As it can be seen, if the decision is sorted and defined by class-hierarchy, the decision formalization is transformed. The following differences can be found between the formalization of the sorted and non-sorted decisions:

- Predicates of the decision options: The first decision case keeps the original $p_1, !p_1$ decision predicates. The result of the first decision case is archived by an instance of one of the subclasses, and it is enclosed by the type of the parent class in the class hierarchy. The archived decision will be reused by the o instanceof $C_{D_{L_1}}$, o instanceof $C_{D_{L_2}}$ predicates on the next decision cases.
- The data-structure which is modified by the decision option will be specified by the keyword 'assignable':
  - The data-structure is the content of the "o" object, which contains the common data structure of the parent class and if the "o" object is the

instance of one of the subclasses, the content is completed with the data structure of the subclass.

- The variables will be referred to in the post-conditions of the decision options (in the subclasses) as it can be seen in the following list:

$$D_{L_1} \Rightarrow ((C_{D_{L_1}})o).v_b, \ldots, ((C_{D_{L_1}})o).v_c$$

$$D_{L_2} \Rightarrow ((C_{D_{L_2}})o).v_e, \ldots, ((C_{D_{L_2}})o).v_f$$

The decision option specific variables of the "o" object – the type of which is the parent class – are achieved by type-forcing.

Accordingly, the object – standing for the parent class in the class hierarchy – encloses the result of the decision. Its decision option specific options can be achieved as already shown.

It is not clear why the usage of type-forcing in the 'assignable' assertions is faulty, but in the post-conditions the usage of type-forcing is required if the data structure of one of the subclasses is required.

4.2.2. More-levels, complex sorted decisions

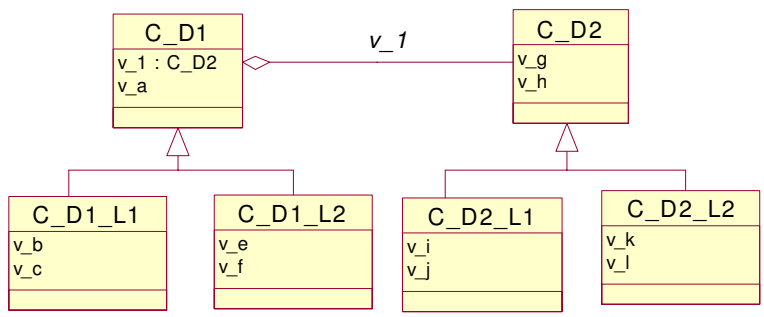

Diagram 2. C_{D_1}, C_{D_2} class-hierarchies by UML diagram [9].

The JML formalization of complex sorted decision on the place of decision-sorting:

/ * @ public normal_behavior //D_2 *in* D_1

@ requires o isntanceof $C_{D_{1L_1}}$; //D_{1L_1} $\Leftrightarrow p_1$

@ {|

@ requires o.v_1 instanceof $C_{D_{2L_1}}$; //D_{2L_1}

@ assignable o; //D_{1L_1}

@ ensures condition$_i$ & & ... & & condition$_o$ & & condition$_p$ & & ... & & condition$_q$ & & //D_{1L_1}

@ condition$_t$ & & ... & & condition$_u$ & & condition$_v$ & & ... & &condition$_w$; //D_{2L_1}

@ also

@ requires o.v_1 instanceof $C_{D_{2L_2}}$; //D_{2L_2}

```
@    assignable o
@    ensures condition1 & & ... & & condition_o & & condition_p & & ... & &condition_q & & //D1L1
@            condition_t & & ... & &conditionl_u & &condition_x & & ... & & condition_y ; //D2L2
@  |}
@ also
@  requires o isntanceof C_D1L2 ; //D1L2  ⇔ !p1
@  {|
@    requires o.v1 instanceof C_D2L1 ; //D2L1
@    assignable o;
@    ensures condition1 & & ... & &condition_o & & condition_r & & ... & & condition_s & & //D1L2
@            conditionl_t & & ... & & condition_u & &condition_v & & ... & &condition_w ; //D2L1
@  also
@    requires o.v1 instanceof C_D2L2 ; //D2L2
@    assignable o;
@    ensures condition1 & & ... & & condition_o & & condition_r & & ... & & condition_s & & //D1L2
@            condition_t & & ... & & condition_u & &condition_x & & ... & &condition_y ; //D2L1
@  |}
@ */
```

4.3. The conditions of well-structured programs based on JML specification

In the following part we describe the facilities of the JML specification of decisions, by which the decision repetitions and the redundant decision definitions can be detected. The full description of these facilities is out of scope of this paper, in the following we just describe the basis of this methodology:

In the Introduction part the following guidelines of a well-structured program were described:

- The methodology and/or the data structure of the decision options have to be defined just once, so the code of the decision options will be defined just once, except it is impossible to sort decisions.

  If the data structures and methodologies of decisions are equivalent, these decisions have to be sorted in the same class-hierarchy. By using JML specification, decisions are equivalent when the data structures of the decisions – which are specified by "assignable" – are equal, and the methodologies of the decisions as the post-conditions of the decision options (specified by "ensures") are equivalent. The decision can be the extension of another one. In this case, one of the data structures is a subset of the other one and there is an implication relation between the postconditions. In this case, the examination of decision predicates is not important.

- Decisions with equivalent decision predicates and different data structures and/or methodologies should not be repeated. In this case, the data structures and the methodologies of the JML specifications of decisions are not equal, but the decision predicates – specified by "requires" – are equivalent. The decision options have to be contracted by sorting them into the same class hierarchy, which will be the common decision abstraction of the contracted decisions. (This case is shown in the Example Code.)

- Decision cases should not be repeated. One decision should be executed just once. (It is the union of the previously mentioned two cases, because the definitions of the decision options are equal, and the decision predicates are equivalent, too.) In this case, the JML formulae of the decisions' data structures and the methodologies are equal and the decision predicates of the decisions (specified by "requires") are equivalent, too.

5. Example

The example shown in this section contains decision-repetition. These decisions have equivalent decision predicates and different data structures, methodologies. According to the previously mentioned conditions of well-structured programs these decisions can be contracted and sorted into class hierarchy, by which the decision-repetition is eliminated.

In the example, the functionality of the purchase is realized: Paying – By Cash/ By Bankcard

The decision about paying mode will be reused later more times. The paying mode determines the parameters, which get as program arguments and it determines the printing data.

The example is based on Java syntax [10].

The two levels of the example code – before and after decision sorting – are also specified by JML, therefore the JML formalization of the two states can be examined and compared.

Accordingly, the Pay class and the Pay class-hierarchy are formalized by JML, by which the differences of the formalizations between the not-sorted and sorted decisions can be described.

5.1. Before decision sorting

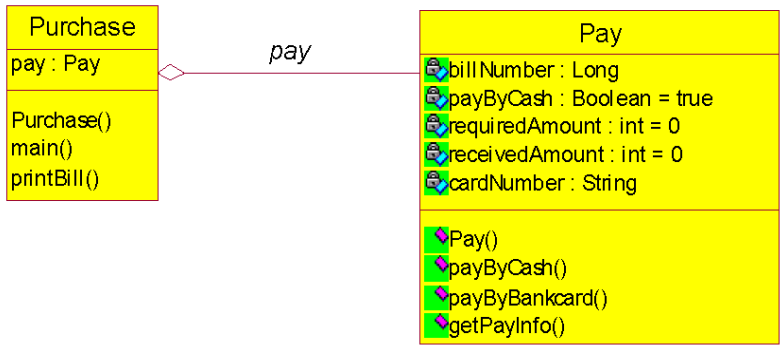

Diagram 3. The two classes of the example before decision sorting by UML diagram [9].

```
package hu.decision.example;
//@ model import org.jmlspecs.models.*;

/** Printing the payment data.
 */
public class Purchase {
  /*@ public static pure model boolean parseable( String s ) {
  @ try { int d = Integer.parseInt(s); return true; }
  @ catch (Exception e) { return false; }}
  @*/

  /*@ public static pure model Pay desidePayingType(String[] args) {
  @    return new Pay(args);
  @}
  @*/

  /**  Payment data - according to the payment type - is got
   *  using the instance of Pay class
   */
  public Pay pay;
  //@ instance invariant pay != null;

  public static void main(String[] args) {
    Purchase purchase=new Purchase();
    purchase.init(args);
    //Printing bill.
    purchase.printBill();
  }

  /** Checking the number of arguments and creating the Pay instance,
   *by which the payment data is printed.
   */
  /*@ private normal_behavior
  @    requires args==null||args.length<4;
  @    assignable \nothing;
  @    ensures false;
  @ also
  @ private normal_behavior
  @    requires args.length>=4&&pay==desidePayingType(args)&&
  @      pay instanceof Pay ;
  @ {|
  @    {|
  @      requires args[0].equals("true")&&parseable(pay.args[1])&&
```

```
  @        parseable(pay.args[2])&&parseable(pay.args[3]);
  @      assignable pay, pay.payByCash, pay.billNumber,
  @        pay.requiredAmount, pay.receivedAmount, System.out;
  @      ensures pay.payByCash==true &&
  @        pay.billNumber==Integer.parseInt(args[1])&&
  @        pay.requiredAmount==Integer.parseInt(args[2]);
  @      ensures pay.receivedAmount==Integer.parseInt(args[3]);
  @    also
  @      requires args[0].equals("false")&&parseable(pay.args[1])&&
  @        parseable(pay.args[2]) && parseable(pay.args[3]);
  @      assignable pay, pay.payByCash, pay.billNumber,
  @         pay.requiredAmount, System.out;
  @      ensures pay.payByCash==false &&
  @        pay.billNumber==Integer.parseInt(args[1])&&
  @        pay.requiredAmount == Integer.parseInt(args[2]);
  @      ensures pay.cardNumber == args[3];
  @    |}
  @ also
  @    requires !parseable(pay.args[1])||!parseable(pay.args[2])||
  @      !parseable(pay.args[3]);
  @    assignable \nothing;
  @    ensures false;
  @ |}
  @*/
  private void init(String[] args)
  {
    //If there are not enough arguments.
    if(args == null || args.length < 4 ){
      System.err.println("There are not enough arguments!");
      System.exit(-1);
    }
    try{
      //Creating the Pay object by which the payment behaviours are realized.
      pay = new Pay(args);
    }
    catch (java.lang.NumberFormatException nfe)
    {
      System.err.println("The format of Arguments is not appropriate!");
      System.exit(-1);
    }

  }

  /** Based on the pay instance payment data is printed.
   */
  /*@ private normal_behavior
  @    requires pay.payByCash==true;
  @    assignable System.out;
  @    ensures (* Prints the Bill Number, Required Amount,
  @        Received Amount*);
  @ also
  @ private normal_behavior
  @    requires pay.payByCash==false;
  @    assignable System.out;
  @    ensures (* Prints the Bill Number, Required Amount,
  @        Card Number*);
  @*/
  private void printBill(){
    String payInfo =pay.getPayInfo();
    System.out.println("Payinfo: "+ payInfo);
  }
}

/*-------------------------------------------------------------------------*/
package hu.decision.example;
//@ model import org.jmlspecs.models.*;

/** Determining payment type (as by cash or by bankcard).
```

```
*/
public  class Pay {

  /*@ public static pure model boolean parseable( String s ) {
  @ try { int d = Integer.parseInt(s); return true; }
  @ catch (Exception e) { return false; }
  @ }
  @*/

  public String[] args;
  //@ invariant args!=null && args.length==4;
  public long billNumber = 0;
  //@ private instance initially billNumber == 0;
  public boolean payByCash=true;
  //@ private instance initially payByCash == true;
  public int requiredAmount=0;
  //@ private instance initially requiredAmount == 0;
  public int receivedAmount=0;
  //@ private instance initially receivedAmount == 0;
  public String cardNumber="";
  //@ private instance initially cardNumber == "";

  /** Determining payment type as by cash or by bankcard.
   *  Getting the bill number and the required amount is
   *  necessary in every case.
   */
  /*@ public behavior
  @ {|
  @   requires args[0].equals("true");
  @   assignable args, payByCash, billNumber, requiredAmount,
  @     receivedAmount;
  @   ensures payByCash==true &&
  @     billNumber==Integer.parseInt(args[1])&&
  @     requiredAmount == Integer.parseInt(args[2]);
  @   ensures receivedAmount == Integer.parseInt(args[3]);
  @ also
  @   requires !parseable(args[1])||!parseable(args[2])||
  @     !parseable(args[3]);
  @   assignable args, payByCash, billNumber, requiredAmount;
  @   ensures false;
  @   signals_only java.lang.NumberFormatException;
  @ |}
  @ also
  @ public behavior
  @ {|
  @   requires args[0].equals("false");
  @   assignable args, payByCash, billNumber, requiredAmount,
  @     receivedAmount, cardNumber;
  @   ensures args==in_args && payByCash==false &&
  @     billNumber==Integer.parseInt(args[1])&&
  @     requiredAmount == Integer.parseInt(args[2]);
  @   ensures cardNumber == args[3];
  @ also
  @   requires !parseable(args[1])||!parseable(args[2]);
  @   assignable args, payByCash, billNumber, requiredAmount;
  @   ensures false;
  @   signals_only java.lang.NumberFormatException;
  @ |}
    @*/
  public Pay(String[] in_args) throws NumberFormatException
  {
    this.args=in_args;
    if(args[0].equals("true"))
      payByCash =true;
    else if(args[0].equals("false"))
      payByCash =false;
    System.out.println("Pay By Cash?:(true/false) "+payByCash);

    billNumber = Integer.parseInt(args[1]);
```

```
    System.out.println("Bill Number:(Number) "+billNumber);

    requiredAmount = Integer.parseInt(args[2]);
    System.out.println("Required Amount:(Number) "+requiredAmount);

    if (payByCash)
      payByCash();
    else
      payByBankcard();
  }

  /** If the customer pays in cash, then getting the
   *  received amount is necessary.
   */
  /*@ private normal_behavior
  @    requires parseable(args[3]);
  @    assignable receivedAmount;
  @    ensures receivedAmount == Integer.parseInt(args[3]);
  @ also
  @ private exceptional_behavior
  @    requires !parseable(args[3]);
  @    assignable receivedAmount;
  @    signals_only java.lang.NumberFormatException;
  @*/
  private void payByCash() throws NumberFormatException
  {
    receivedAmount = Integer.parseInt(args[3]);
    System.out.println("Received Amount:(Number) "+receivedAmount);
  }

  /** If the customer pays by bankcard, then getting
   *  the card-number is necessary.
   */
  /*@ private normal_behavior
  @ assignable  cardNumber;
  @ ensures  cardNumber == args[3];
  @*/
  private void payByBankcard()
  {
    cardNumber = args[3];
    System.out.println("cardNumber:(String)"+cardNumber);
  }

  /** Printing payment data according to payment type.
   */
  /*@ public normal_behavior
  @    requires payByCash == true;
  @    assignable \nothing;
  @    ensures \result == "Bill Number: "+String.valueOf(billNumber)+
  @      "; Required Amount: "+String.valueOf(requiredAmount)+
  @      "; Received Amount: "+String.valueOf(receivedAmount);
  @ also
  @ public normal_behavior
  @    requires payByCash == false;
  @    assignable \nothing;
  @    ensures \result == "Bill Number: "+String.valueOf(billNumber)+
  @      "; Required Amount: "+String.valueOf(requiredAmount)+
  @      "; Card Amount: "+String.valueOf(cardNumber);
  @*/
  public String getPayInfo()
  {
    if (payByCash)
      return "Bill Number: "+String.valueOf(billNumber)+
      "; Required Amount: "+String.valueOf(requiredAmount)+
      "; Received Amount: "+String.valueOf(receivedAmount);
    else
      return "Bill Number: " + String.valueOf(billNumber)+
      "; Required Amount: "+ String.valueOf(requiredAmount)+
```

```
        "; Card Number: " + String.valueOf(cardNumber);
    }
}
```

The decision predicate of the decision about getting paying data is realized in the Pay constructor as follows:

```
    @ requires args[0].equals("true");
      @   …
      @ also
      @   requires args[0].equals("false");
      @   …
```

The decision predicate of printing data decision in the getPayInfo method is equivalent with the predicate of the decision about getting paying data:

```
    @ requires payByCash == true;
      @   …
      @ also
      @   requires payByCash == false;
      @   …
```

The decision predicate of the decision about printing data (payByCash variable) is evaluated in the decision options of the other decision (about getting paying data) based on its decision predicate (`args[0].equals("true")`). Therefore the two decision predicates are eqivalent, accordingly the two decisions can be contracted sorting them into the same class hierarchy.

5.2. After decision sorting

The decisions about payment type are sorted into the class hierarchy, where the different paying modes are defined in the subclasses as the decision options. If somebody pays in cash, the number of the bankcard and the transaction number are not required, but the paid and received amounts are required. In case of paying by bankcard, the received and paid amounts are not required, but the bankcard number and the transaction number are needed. After the executing the contraction of the equivalent decisions of the paying mode (which were in the 'Pay' and the 'getPayInfo' methods), the decision about paying mode will be executed just once. This will be enclosed and archived by the 'Pay' class hierarchy and the enclosed decision will be reused on the next occasions.

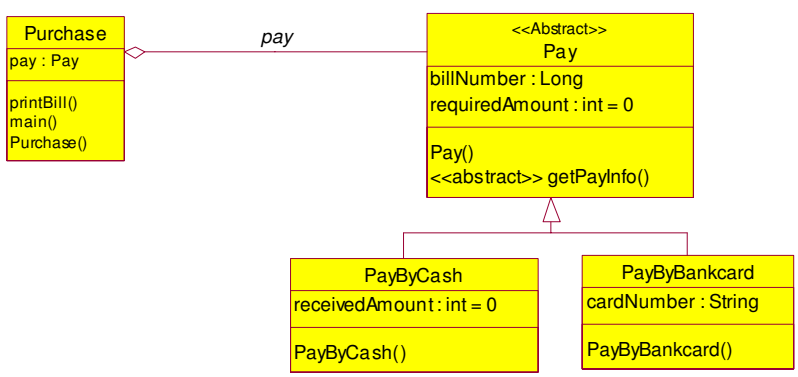

Diagram 4. Classes of the example after decision sorting by UML diagram [9].

```
package hu.decision.example2;
//@ model import org.jmlspecs.models.*;

/** Printing the payment data.
 */
public class Purchase {

  /*@ public static pure model boolean parseable( String s ) {
  @ try { int d = Integer.parseInt(s); return true; }
  @ catch (Exception e) { return false; }}
  @*/

  /*@ public static pure model Pay desidePayingType(String[] args) {
  @ if(args[0].equals("true")||(!args[0].equals("true")&&
    @    !args[0].equals("false")))
  @   return new PayByCash(args);
  @ else
  @   return new PayByBankcard(args);
  @}
  @*/

  /**  Payment data - according to the payment type - is got
   *  using the instance of PayByCash or PayByBankcard class
   */
  public Pay pay;
  //@ instance invariant pay != null;

  public static void main(String[] args) {
      Purchase purchase=new Purchase();
      purchase.init(args);
      purchase.printBill();
  }

  /** Checking the number of arguments and creating the instance
   *  of PayByCash or PayByBankcard class, by which the payment data
   *  is printed.
   */
  /*@
  @ private normal_behavior
  @   requires args==null||args.length<4;
  @   assignable \nothing;
  @   ensures false;
  @ also
  @ private normal_behavior
  @   requires args!=null&&args.length>=4&&pay==desidePayingType(args);
  @ {|
  @     {|
  @     requires pay instanceof PayByCash;
  @     assignable pay, System.out;
  @     ensures pay.billNumber==Integer.parseInt(pay.args[1])&&
  @       pay.requiredAmount==Integer.parseInt(pay.args[2]);
  @     ensures ((PayByCash)pay).receivedAmount==Integer.parseInt(pay.args[3]);
  @   also
  @     requires pay instanceof PayByBankcard;
  @     assignable pay, System.out;
  @     ensures pay.billNumber==Integer.parseInt(pay.args[1])&&
  @       pay.requiredAmount==Integer.parseInt(pay.args[2]);
  @     ensures ((PayByBankcard)pay).cardNumber==pay.args[3];
  @   |}
  @ also
  @   requires !parseable(args[1])||!parseable(args[2])||
  @     !parseable(args[3]);
  @   assignable \nothing;
  @   ensures false;
  @ |}
  @*/
  private void init(String[] args){
    //If there are not enough arguments.
```

```
    if(args == null || args.length < 4 ){
      System.err.println("There are not enough arguments!");
      System.exit(-1);
    }
    try{
      //Creating the Pay object by which the payment behaviours
      //are realized.
      if(args[0].equals("true"))
        pay=new PayByCash(args);
      else if(args[0].equals("false"))
        pay=new PayByBankcard(args);
      else
        pay=new PayByCash(args);
      System.out.println("PayByCash?:(true/false) "+args[0]);
    }
    catch (java.lang.NumberFormatException nfe){
      System.err.println("The format of the Arguments is not appropriate!");
      System.exit(-1);
    }
  }

  /** Based on the pay instance the payment data is printed.
   */
  /*@ private normal_behavior
  @   requires pay instanceof PayByCash;
  @   assignable System.out;
  @   ensures (* Prints the Bill Number, Required Amount,
  @     Received Amount*);
  @ also
  @ private normal_behavior
  @   requires pay instanceof PayByBankcard;
  @   assignable System.out;
  @   ensures (* Prints the Bill Number, Required Amount,
  @     Card Number*);
  @*/
  private void printBill(){
    String payInfo = pay.getPayInfo();
    System.out.println("Payinfo: "+ payInfo);
  }
}

/*-------------------------------------------------------------------------*/
package hu.decision.example2;
//@ model import org.jmlspecs.models.*;

/** The parent class of payment type class hierarcy.
 *It determines the common data structure and the behaviour
 *of the subclasses (payment types).
 */
public abstract class Pay {

  /*@ public static pure model boolean parseable( String s ) {
  @ try { int d = Integer.parseInt(s); return true; }
  @ catch (Exception e) { return false; }}
  @*/
  protected /*@ spec_public @*/ String[] args;
  //@ invariant args!=null && args.length==4;
  protected /*@ spec_public @*/ long billNumber = 0;
  //@ private instance initially billNumber == 0;
  protected /*@ spec_public @*/ int requiredAmount=0;
  //@ private instance initially requiredAmount == 0;

  /**
   *Getting the bill number and the required amount, which are the
   *common data structure of payment types.
   */
  /*@ public behavior
  @   requires parseable(args[1])&&parseable(args[2]);
```

```
  @    assignable args, billNumber, requiredAmount, System.out;
  @    ensures args==in_args && billNumber==Integer.parseInt(args[1])&&
  @    requiredAmount==Integer.parseInt(args[2]);
  @ also
  @    requires !parseable(args[1])||!parseable(args[2]);
  @    assignable args;
  @    ensures args == in_args;
  @    signals_only java.lang.NumberFormatException;
  @*/
  public Pay(String[] in_args) throws NumberFormatException
  {
    this.args=in_args;
    billNumber = Integer.parseInt(args[1]);
    System.out.println("Bill Number:(Number) "+billNumber);
    requiredAmount = Integer.parseInt(args[2]);
    System.out.println("Required Amount:(Number) "+requiredAmount);
  }

  /** Getting the payment data according to payment type. The behaviour
   *    is realized by the subclasses of the Pay class.
   */
  abstract public String getPayInfo();
}


/*-------------------------------------------------------------------------*/
package hu.decision.example2;
//@ model import org.jmlspecs.models.*;

/**
 * The PayByBankcard class as the subclass of the Pay class is available,
 * if the customer pays by bankcard as it is decided in the Main method.
 */
public class PayByBankcard extends Pay{

  public  String cardNumber;

  /** If the customer pays by bankcard,
   *  then getting the card-number is necessary.
   */
  /*@ also
  @ public behavior
  @    requires parseable(args[3]);
  @    assignable cardNumber, System.out;
  @    ensures cardNumber==Integer.parseInt(args[3]);
  @ also
  @    requires !parseable(args[1])||!parseable(args[2]);
  @    assignable args;
  @    ensures args == in_args;
  @    signals_only java.lang.NumberFormatException;
  @*/
  public PayByBankcard(String[] args)
  { super(args);
    cardNumber = args[3];
    System.out.println("cardNumber Amount:(String)"+cardNumber);
  }

  /** Printing the payment data according to the payment type.
   */
  /*@ public normal_behavior
  @    assignable \nothing;
  @    ensures \result == "Bill Number: "+String.valueOf(billNumber)+
  @    "; Required Amount: "+String.valueOf(requiredAmount)+
  @    "; Card Number: "+String.valueOf(cardNumber);
  @*/
  public String getPayInfo()
  {
    return "Bill Number: " + String.valueOf(billNumber)+
```

```
          "; Required Amount: "+ String.valueOf(requiredAmount)+
          "; Card Number: " + String.valueOf(cardNumber);
    }
}

/*-------------------------------------------------------------------*/
package hu.decision.example2;
//@ model import org.jmlspecs.models.*;

/**
 * The PayByCash class as the subclass of the Pay class is available,
 * if the customer pays in cash as it is decided in the Main method.
 */
public class PayByCash extends Pay{

  protected /*@ spec_public @*/ int receivedAmount=0;
  //@ public instance initially receivedAmount == 0;

  /** If the customer pays in cash,
   *  then getting the received amount is necessary.
   */
    /*@ public normal_behavior
  @    requires parseable(args[3]);
  @    assignable receivedAmount ,System.out;
  @    ensures receivedAmount == Integer.parseInt(args[3]);
  @ also
  @ public exceptional_behavior
  @    requires !parseable(args[3]);
  @    assignable receivedAmount, System.out;
  @    signals_only java.lang.NumberFormatException;
  @*/
  public PayByCash(String[] args) throws NumberFormatException
  { super(args);
    receivedAmount = Integer.parseInt(args[3]);
    System.out.println("Received Amount:(Number) "+receivedAmount);
  }

  /** Printing the payment data according to the payment type.
   */
  /*@ public normal_behavior
  @    assignable \nothing;
  @    ensures \result == "Bill Number: "+String.valueOf(billNumber)+
  @       "; Required Amount: "+String.valueOf(requiredAmount)+
  @       "; Received Amount: "+String.valueOf(receivedAmount);
  @*/
  public String getPayInfo()
  {
    return   "Bill Number: "+String.valueOf(billNumber)+
         "; Required Amount: "+String.valueOf(requiredAmount)+
         "; Received Amount: "+String.valueOf(receivedAmount);
  }
}
```

The decisions about paying mode with different methodologies will be defined in the Pay class hierarchy. The two decision options differ in receiving and printing data about paying.

The PayByCash class – as the subclass of the Pay class – is available, if the customer pays by cash as it is decided in the Main method.

The PayByBankcard class – as the subclass of the Pay class – is available, if the customer pays by bankcard as it is decided in the Main method.

The instantiation can be found in the ‘init’ method, by which the decision can be enclosed and archived by sorting it referring to an aggregation as a variable (pay

object). The archived decision can be used in the next decision occasions without knowing about the result of the decision.

```
if(args[0].equals("true"))
  pay=new PayByCash(args);
else if(args[0].equals("false"))
  pay=new PayByBankcard(args);
else
  pay=new PayByCash(args);
```

The JML foramlization of enlosing:

```
  @ public static pure model Pay desidePayingType(String[] args) {
  @ if(args[0].equals("true")||(!args[0].equals("true")&&
  @    !args[0].equals("false")))
  @    return new PayByCash(args);
  @ else
  @    return new PayByBankcard(args);
  @}
...
  @    requires args!=null&&args.length>=4&&pay==desidePayingType(args);
...
```

The archived decision can be reused in the next decision cases based on pay object as follows:

```
  @      requires pay instanceof PayByCash;
  @      ...
  @    also
  @      requires pay instanceof PayByBankcard;
  @      ...
```

The type of the pay object determines the appropriate decision option for the next decision occasions, accordingly the decision enclosing is realized.

6. Conclusion

The new interpretation of inheritance – as an extension of the old interpretation – is introduced, and described by an example. Accordingly, the aim of the application of inheritance and the object-oriented paradigms is the elimination of decision repetition by sorting the decisions' definitions into class hierarchy. By using the object-oriented paradigms, the consistence of the decisions can be solved making the maintenance of the program easier.

In the Introduction, we showed the properties of well-structured programs. In order that these properties could be examined, the formalization of the decisions is introduced by JML. Based on JML, the non-sorted and sorted states of the decisions can be described realizing the formal differences between them.

We have used the JML formalization method in order to examine the cases of decision repetitions and the relations of complex decisions.

As it was mentioned in [11], there are connections between the decision based interpretation of object-oriented paradigms and Design Patterns, accordingly Design Pattern gives us recipes to eliminate decision redundancy and to archive decisions. As JML is adapted to examine the decisions and the decision repetitions of object-oriented programs – as it was mentioned in this paper – we think JML is adapted to formalize Design Patterns more exactly than the existing formalization tools.

As for the idea – which was introduced by [11] and examined in this paper by

JML formalization – was created in the course of analyzing of Design Patterns, we intend to examine Design Patterns based on JML formalization, and to examine the additional connections between the applicability of Design Patterns and decision repetitions

Based on the new decision-based conception, we can realize more manifest and exact explanations for the aims of Design Patterns. By using the new idea, a new, more natural classification of Design Patterns is described in [11], by which we would like to launch a discussion about a new interpretation of the existing classification [8].

According to our plan, we will examine whether the decision repetition in the design and the source can be eliminated by automatic sorting, that helps to upgrade the quality of the design and the source automatically.

References

[1] Booch, G., Object-Oriented Analysis and Design with Applications, *Adison-Wesley*, 1994.

[2] Brito e Abreu, F., Melo, W., Evaluating the Impact of Object-Oriented Design on Software Quality - *Originally published in Proceedings of the 3rd International Software Metrics Symposium (METRICS'96)*, IEEE, Berlin, Germany, 1996.

[3] Piefel, M., Object-Oriented Software Development - *Coursework 'Information Engineering', Department of Computing, University of Bradford*, 1996/97.

[4] Software Quality Metrics for Object-Oriented System Environments, *Software Assurance Technology Center as SATC*, 1995.

[5] Nierstrasz, O., Survey of Object-Oriented Concepts, *University of Geneva*.

[6] Fisher, K., C. Mitchell, J., Notes on typed object-oriented programming, *Computer Science Dept.*, *Stanford University, Stanford*, 1994.

[7] Moore, I., Automatic Inheritance Hierarchy Restructuring and Method Refactoring, *Conference on Object-Oriented Programming Systems Languages and Applications San Jose*, California, United States, 1996.

[8] Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, *Addison-Wesley Professional Computing Series*, 1995.

[9] Rambaugh, J., Jacobson, I., Booch, G., The unified modeling language reference manual, *Addision-Wesley*, 1998.

[10] JavaTM 2 Platform Standard Edition, `http://java.sun.com/j2se/1.4.2/docs`

[11] Márien, Sz., Decision Based Examination of Object-Oriented Programming and Design Patterns, *Teaching Mathematics and Computer Science*, Debrecen, Hungary, 2008.

[12] Burdy, L., Cheon, Y., Cok, D., Ernst, M.D., Kiniry, J.R., Leavens, G.T., Leino, K., Rustan, M., Poll, E., An overview of JML tools and applications, 2004.

[13] Leavens, G.T., L. Baker A., Ruby, C., JML: A Notation for Detailed Design, 1999.

[14] Chalin, P., Kiniry, J.R., Leavens, G.T., Poll, E., Beyond Assertions: Advanced Specification and Verification with JML and ESC/Java2.

[15] Leavens, G.T., Baker, A.L., Ruby, C., Preliminary Design of JML: A Behavioural Interface Specification Language for Java, 2006.

Szabolcs Márien
University of Debrecen
Debrecen, Hungary
e-mail: mariensz@hotmail.com

Annales Mathematicae et Informaticae
***35** (2008) pp. 123–128*
http://www.ektf.hu/ami

Further generalizations of the Fibonacci-coefficient polynomials*

Ferenc Mátyás

Institute of Mathematics and Informatics
Eszterházy Károly College, Eger, Hungary

Submitted 11 September 2008; Accepted 10 November 2008

Abstract

The aim of this paper is to investigate the zeros of the general polynomials

$$q_n^{(i,t)}(x) = \sum_{k=0}^{n} R_{i+kt}x^{n-k} = R_i x^n + R_{i+t}x^{n-1} + \cdots + R_{i+(n-1)t}x + R_{i+nt},$$

where $i \geqslant 1$ and $t \geqslant 1$ are fixed integers.

Keywords: Second order linear recurrences, bounds for zeros of polynomials with special coefficients

MSC: 11C08, 13B25

1. Introduction

The the second order linear recursive sequence

$$R = \{R_n\}_{n=0}^{\infty}$$

is defined by the following manner: let $R_0 = 0$, $R_1 = 1$, A and B be fixed positive integers. Then for $n \geqslant 2$

$$R_n = AR_{n-1} + BR_{n-2}. \tag{1.1}$$

According to the known Binet-formula, for $n \geqslant 0$

$$R_n = \frac{\alpha^n - \beta^n}{\alpha - \beta},$$

*Research has been supported by the Hungarian OTKA Foundation No. T048945.

where α and β are the zeros of the characteristic polynomial $x^2 - Ax - B$ of the sequence R. We can suppose that $\alpha > 0$ and $\beta < 0$.

In the special case $A = B = 1$ we can get the wellknown Fibonacci-sequence, that is, with the usual notation

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2} \quad (n \geqslant 2).$$

According to D. Garth, D. Mills and P. Mitchell [1] the definition of the Fibonacci-coefficient polynomials $p_n(x)$ is the following:

$$p_n(x) = \sum_{k=0}^{n} F_{k+1} x^{n-k} = F_1 x^n + F_2 x^{n-1} + \cdots + F_n x + F_{n+1}.$$

In [3] we delt the zeros of the polynomials $q_n(x)$, where

$$q_n(x) = \sum_{k=0}^{n} R_{k+1} x^{n-k} = R_1 x^n + R_2 x^{n-1} + \cdots + R_n x + R_{n+1},$$

that is, our results concerned to a family of the linear recursive sequences of second order.

The aim of this revisit of the theme is to investigate the zeros of the much more general polynomials $q_n^{(i)}(x)$ and $q_n^{(i,t)}(x)$, where $i \geqslant 1$ and $t \geqslant 1$ are fixed integers:

$$q_n^{(i)}(x) = \sum_{k=0}^{n} R_{i+k} x^{n-k} = R_i x^n + R_{i+1} x^{n-1} + \cdots + R_{i+n-1} x + R_{i+n}, \qquad (1.2)$$

$$q_n^{(i,t)}(x) = \sum_{k=0}^{n} R_{i+kt} x^{n-k} = R_i x^n + R_{i+t} x^{n-1} + R_{i+2t} x^{n-2} \cdots + R_{i+(n-1)t} x + R_{i+nt}.$$

2. Preliminary and known results

At first we mention that the polynomials $q_n^{(i)}(x)$ can easily be rewritten in a recursive manner. That is, if $q_0^{(i)}(x) = R_i$ then for $n \geqslant 1$

$$q_n^{(i)}(x) = x q_{n-1}^{(i)}(x) + R_{i+n}.$$

We need the following three lemmas:

Lemma 2.1. *For $n \geqslant 1$ let $g_n^{(i)}(x) = (x^2 - Ax - B) q_n^{(i)}(x)$. Then*

$$g_n^{(i)}(x) = R_i x^{n+2} + B R_{i-1} x^{n+1} - R_{i+n+1} x - B R_{i+n}.$$

Proof. Using (1.2) we get $q_1^{(i)}(x) = R_i x + R_{i+1}$ and by (1.1) $g_1^{(i)}(x) = (x^2 - Ax - B) q_1^{(i)}(x) = (x^2 - Ax - B)(R_i x + R_{i+1}) = \cdots = R_i x^3 + B R_{i-1} x^2 - R_{i+2} x - B R_{i+1}$.

Continuing the proof with induction on n, we suppose that the statement is true for $n-1$ and we prove it for n. Applying (1.2) and (1.1), after some numerical calculations one can get that

$$\begin{aligned} g_n^{(i)}(x) &= (x^2 - Ax - B)q_n^{(i)}(x) \\ &= xg_{n-1}^{(i)}(x) + (x^2 - Ax - B)R_{i+n} = \cdots \\ &= R_i x^{n+2} + BR_{i-1}x^{n+1} - R_{i+n+1}x - BR_{i+n}. \end{aligned}$$

□

Lemma 2.2. *If every coefficients of the polynomial* $f(x) = a_0 + a_1x + \cdots + a_nx^n$ *are positive numbers and the roots of equation* $f(x) = 0$ *are denoted by* $z_1, z_2, \ldots, z_n$, *then*

$$\gamma \leqslant |z_i| \leqslant \delta$$

hold for every $1 \leqslant i \leqslant n$, *where* γ *is the minimal, while* δ *is the maximal value in the sequence*

$$\frac{a_0}{a_1}, \frac{a_1}{a_2}, \ldots, \frac{a_{n-1}}{a_n}.$$

Proof. Lemma 2.2 is known as theorem of S. Kakeya [4]. □

Lemma 2.3. *Let us consider the sequence* R *defined by* (1.1). *The increasing order of the elements of the set*

$$\left\{ \frac{R_{j+1}}{R_j} \ : \ 1 \leqslant j \leqslant n \right\}$$

is

$$\frac{R_2}{R_1}, \frac{R_4}{R_3}, \frac{R_6}{R_5}, \ldots, \frac{R_7}{R_6}, \frac{R_5}{R_4}, \frac{R_3}{R_2}.$$

Proof. Lemma 2.3 can be found in [2]. □

3. Results and proofs

At first we deal with the number of the real zeros of the polynomial $q_n^{(i)}(x)$ defined in (1.2), that is

$$q_n(^{(i)}x) = \sum_{k=0}^{n} R_{i+k}x^{n-k} = R_ix^n + R_{i+1}x^{n-1} + \cdots + R_{i+n-1}x + R_{i+n}.$$

Theorem 3.1. a) *If* $n \geqslant 2$ *and even, then the polynomial* $q_n^{(1)}(x)$ *has not any real zero, while if* $i \geqslant 2$ *then* $q_n^{(i)}(x)$ *has no one or has two negative real zeros, that is, every zeros – except at most two – are non-real complex numbers.*

b) *If* $n \geqslant 3$ *and odd, then the polynomial* $q_n^{(i)}(x)$ *has only one real zero and this is negative. That is, every but one zeros are non-real complex numbers.*

Proof. Because of the definition (1.1) of the sequence R the coefficients of the polynomials $q_n^{(i)}(x)$ are positive ones, thus positive real root of the equation $q_n^{(i)}(x) = 0$ does not exist. That is, it is enough to deal with only the existence of negative roots of the equation $q_n^{(i)}(x) = 0$. a) Since n is even, the coefficients of the polynomial

$$\begin{aligned} g_n^{(i)}(-x) &= R_i(-x)^{n+2} + BR_{i-1}(-x)^{n+1} - R_{i+n-1}(-x) - BR_{i+n} \\ &= R_i x^{n+2} - BR_{i-1}x^{n+1} + R_{i+n-1}x - BR_{i+n} \end{aligned}$$

has only one change of sign if $i = 1$, thus according to the Descartes' rule of signs, the polynomial $g_n^{(i)}(x)$ has exactly one negative real zero. But $g_n^{(i)}(x) = (x^2 - Ax - B)q_n^{(i)}(x)$ implies that $g_n^{(i)}(\beta) = 0$, where $\beta < 0$, and so the polynomial $q_n^{(i)}(x)$ can not have any negative real zero if $i = 1$. But in the case $i \geqslant 2$ the polynomial $g_n^{(i)}(-x)$ has 3 changes of sign, that is, $q_n^{(i)}(x) = 0$ has no one or 2 negative roots.

b) Since $n \geqslant 3$ is odd, thus the existence of at least one negative real zero is obvious. We have only to prove that exactly one negative real zero exists. The polynomial

$$\begin{aligned} g_n^{(i)}(-x) &= R_i(-x)^{n+2} + BR_{i-1}(-x)^{n+1} - R_{i+n-1}(-x) - BR_{i+n} \\ &= -R_i x^{n+2} + BR_{i-1}x^{n+1} + R_{i+n-1}x - BR_{i+n} \end{aligned}$$

shows that among its coefficients there are two changes of signs, thus according to the Descartes' rule of signs, the polynomial $g_n^{(i)}(x)$ has either two negative real zeros or no one. But $g_n^{(i)}(x) = (x^2 - Ax - B)q_n^{(i)}(x)$ implies that for $\beta < 0$ $g_n^{(i)}(\beta) = 0$. Although, $g_n^{(i)}(\alpha) = 0$ also holds, but $\alpha > 0$. That is, an other negative real zero of $g_n^{(i)}(x)$ must exist. Because of $g_n^{(i)}(x) = (x^2 - Ax - B)q_n^{(i)}(x)$ this zero must be the zero of the polynomial $q_n^{(i)}(x)$.

This terminated the proof of the theorem. □

Remark 3.2. Some numerical examples imply the conjection that if n is even and $i \geqslant 2$ then $q_n^{(i)}(x)$ has no negative real root.

In the following part of this note we deal with the localization of the zeros of the polynomials

$$q_n^{(i)}(x) = \sum_{k=0}^{n} R_{i+k}x^{n-k} = R_i x^n + R_{i+1}x^{n-1} + \cdots + R_{i+n-1}x + R_{i+n}.$$

Theorem 3.3. *Let $z \in \mathbb{C}$ denote an arbitrary zero of the polynomial $q_n^{(i)}(x)$ if $n \geqslant 1$. Then*

$$\frac{R_{i+1}}{R_i} \leqslant |z| \leqslant \frac{R_{i+2}}{R_{i+1}},$$

if i is odd, while

$$\frac{R_{i+2}}{R_{i+1}} \leqslant |z| \leqslant \frac{R_{i+1}}{R_i},$$

if i is even.

Proof. To apply Lemma 2.2 for the polynomial $q_n^{(i)}(x)$ we have to determine the minimal and maximal values in the sequence

$$\frac{R_{i+n}}{R_{i+n-1}}, \frac{R_{i+n-1}}{R_{i+n-2}}, \ldots, \frac{R_{i+1}}{R_i}.$$

Applying Lemma 2.3, one can get the above stated bounds. □

Remark 3.4. Even more there is an other possibility for further generalization. Let $i \geqslant 1$ and $t \geqslant 1$ be fixed integers.

$$q_n^{(i,t)}(x) := \sum_{k=0}^{n} R_{i+kt}x^{n-k} = R_i x^n + R_{i+t}x^{n-1} + R_{i+2t}x^{n-2} \cdots + R_{i+(n-1)t}x + R_{i+nt}.$$

The following recursive relation also holds if $q_0^{(i,t)}(x) = R_i$ then for $n \geqslant 1$

$$q_n^{(i,t)}(x) = xq_{n-1}^{(i,t)}(x) + R_{i+nt}.$$

Using similar methods for the set

$$\left\{ \frac{R_{i+jt}}{R_{i+(j-1)t}} \ : \ 1 \leqslant j \leqslant n \right\}$$

it can be proven that for any zero z of $q_n^{(i,t)}(x) = 0$:
if i and t are odd then:

$$\frac{R_{i+t}}{R_i} \leqslant |z| \leqslant \frac{R_{i+2t}}{R_{i+t}},$$

if i is even and t is odd then:

$$\frac{R_{i+2t}}{R_{i+t}} \leqslant |z| \leqslant \frac{R_{i+t}}{R_i},$$

if i and t are even then:

$$\frac{R_{i+nt}}{R_{i+(n-1)t}} \leqslant |z| \leqslant \frac{R_{i+t}}{R_i},$$

if i is odd and t is even then:

$$\frac{R_{i+t}}{R_i} \leqslant |z| \leqslant \frac{R_{i+nt}}{R_{i+(n-1)t}}.$$

References

[1] Garth, D., Mills, D, Mitchell, P., Polynomials generated by the fibonacci sequence, *Journal of Integer Sequences*, Vol. 10 (2007) Article 07.6.8.

[2] Mátyás, F., On the quotions of the elements of linear recursive sequences of second order, *Matematikai Lapok*, 27 (1976–1979), 379–389. (Hungarian).

[3] Mátyás, F., On the generalization of the Fibonacci-coefficient polynomials, *Annales Mathematicae et Informaticae*, 34 (2007) 71–75.

[4] Zemyan, S.M., On the zeros of the n-th partial sum of the exponential series, *The American Mathematical Monthly*, 112 (2005) No. 10, 891–909.

Ferenc Mátyás
Institute of Mathematics and Informatics
Eszterházy Károly College
P.O. Box 43
H-3301 Eger
Hungary
e-mail: matyas@ektf.hu

Annales Mathematicae et Informaticae
***35** (2008) pp. 129–134*
http://www.ektf.hu/ami

Generalization of some inequalities for the q-gamma function

Armend Sh. Shabani

Department of Mathematics, University of Prishtina

Submitted 25 May 2008; Accepted 5 September 2008

Abstract

In this paper is obtained q-analogue of a double inequality involving the Euler's gamma function proved in [5]. In the same way, the paper [5] generalized papers [1]–[4], this paper will generalize some inequalities for the q-gamma function such as those presented in [9, 10].

Keywords: q-gamma function, Inequalities

MSC: 33D05

1. Introduction

The Euler gamma function $\Gamma(x)$ is defined for $x > 0$ by

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt.$$

The Psi or digamma function, the logarithmic derivative of the gamma function is defined by

$$\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}, x > 0.$$

The q-analogue of the gamma function is defined by

$$\Gamma_q(x) = (1-q)^{1-x} \prod_{i=1}^{\infty} \frac{1-q^i}{1-q^{x+i}}, \quad q \in (0,1). \tag{1.1}$$

The q-psi function is defined as

$$\psi_q(x) = \frac{d}{dx} \log \Gamma_q(x). \tag{1.2}$$

We will make use of the following well known facts

$$\lim_{q\to 1^-}\Gamma_q(x)=\Gamma(x),\ \lim_{q\to 1^-}\psi_q(x)=\psi(x). \tag{1.3}$$

R. Askey, [8] derived some properties of the q-gamma function.

Papers [1, 2, 3, 4] were related to some double inequalities involving the gamma function.

In [5] the following theorem is proved:

Theorem 1.1. *Let f be a function defined by*

$$f(x)=\frac{\Gamma(a+bx)^c}{\Gamma(d+ex)^f},\quad x\geqslant 0, \tag{1.4}$$

where a,b,c,d,e,f are real numbers such that: $a+bx>0, d+ex>0, a+bx\leqslant d+ex$.

In both situations:

i) *Let $ef\geqslant bc>0$. If $\psi(a+bx)>0$ or $\psi(d+ex)>0$*

ii) *Let $bc\geqslant ef>0$. If $\psi(d+ex)<0$ or $\psi(a+bx)<0$*

the function f is decreasing for $x\geqslant 0$ and for $x\in[0,1]$ the following double inequality holds:

$$\frac{\Gamma(a+b)^c}{\Gamma(d+e)^f}\leqslant\frac{\Gamma(a+bx)^c}{\Gamma(d+ex)^f}\leqslant\frac{\Gamma(a)^c}{\Gamma(d)^f}. \tag{1.5}$$

which represents a generalization of inequalities given in [1, 2, 3, 4].

Some of those inequalities were generalized using q-gamma analogue function. Thus T. Kim and C. Adiga [9] proved:

Theorem 1.2. *If $0<q<1, a\geqslant 1$ and $x\in[0,1]$ then*

$$\frac{1}{\Gamma_q(1+a)}\leqslant\frac{\Gamma_q(1+x)^a}{\Gamma_q(1+ax)}\leqslant 1. \tag{1.6}$$

Letting q tend to 1 and $a=n$, one obtains q-gamma analogue to the inequality given in [1]. Letting q tend to 1, one obtains q-gamma analogue to the inequality given in [2].

Recently, T. Mansour [10] proved:

Theorem 1.3. *Let $x\in[0,1]$, $q\in(0,1)$, $a\geqslant b>0$, c, d positive real numbers with $bc\geqslant ad$ and $\psi_q(b+ax)>0$ then*

$$\frac{\Gamma_q(a)^c}{\Gamma_q(b)^d}\leqslant\frac{\Gamma_q(a+bx)^c}{\Gamma_q(b+ax)^d}\leqslant\frac{\Gamma_q(a+b)^c}{\Gamma_q(a+b)^d}. \tag{1.7}$$

which again by letting q to 1 gives q-gamma analogue of inequality given in [4] and thus gives a generalization of the main results of [4].

The idea of this paper is to consider the q-gamma analogue of the function given by Theorem 1.1, so to consider the function:

$$f(x) = \frac{\Gamma_q(a+bx)^c}{\Gamma_q(d+ex)^f}, \quad x \geqslant 0 \tag{1.8}$$

and to have q-analogue results of [5] and thus to generalize the results of [9] and [10].

2. Results

In order to establish the proof of the theorems, we need the following lemmas:

Lemma 2.1. *The q-psi function has the following series representation:*

$$\psi_q(x) = -\log(1-q) + \log q \cdot \sum_{i=0}^{\infty} \frac{q^{x+i}}{1-q^{x+i}}. \tag{2.1}$$

Proof. See [7]. □

Lemma 2.2. *Let $q \in (0,1)$, $x > 0$, $y > 0$ and $x < y$. Then*

$$\psi_q(x) < \psi_q(y). \tag{2.2}$$

Proof. Using Lemma 2.1 we obtain:

$$\begin{aligned}
\psi_q(x) - \psi_q(y) &= \log q \cdot \Big(\sum_{i=0}^{\infty} \frac{q^{x+i}}{1-q^{x+i}} - \sum_{i=0}^{\infty} \frac{q^{y+i}}{1-q^{y+i}} \Big) \\
&= \log q \cdot \sum_{i=0}^{\infty} \Big(\frac{q^{x+i}}{1-q^{x+i}} - \frac{q^{y+i}}{1-q^{y+i}} \Big) \\
&= \log q \cdot \sum_{i=0}^{\infty} \frac{q^{x+i} - q^{y+i}}{(1-q^{x+i})(1-q^{y+i})} \\
&= \log q \cdot \sum_{i=0}^{\infty} \frac{q^i(q^x - q^y)}{(1-q^{x+i})(1-q^{y+i})} < 0,
\end{aligned}$$

because for $x < y$ and $q \in (0,1)$ we have $q^x > q^y$ and $\log q < 0$ which completes the proof. □

Lemma 2.3. *Let $q \in (0,1)$, $a + bx > 0$, $d + ex > 0$ and $a + bx \leqslant d + ex$. Then*

$$\psi_q(a+bx) - \psi_q(d+ex) \leqslant 0. \tag{2.3}$$

Proof. By Lemma 2.2. □

Lemma 2.4. *Let a, b, c, d, e, f be real numbers such that $a + bx > 0$, $d + ex > 0$, $a + bx \leqslant d + ex$ and $ef \geqslant bc > 0$. Let $q \in (0, 1)$. If*

(i) $\psi_q(a + bx) > 0$ *or*

(ii) $\psi_q(d + ex) > 0$

then

$$bc\psi_q(a + bx) - ef\psi_q(d + ex) \leqslant 0. \tag{2.4}$$

Proof. (i) Let $\psi_q(a+bx) > 0$. From Lemma 2.3 we have $\psi_q(d+ex) \geqslant \psi_q(a+bx) > 0$. Multiplying both sides of inequality $ef \geqslant bc$ with $\psi_q(d + ex)$ we obtain

$$ef\psi_q(d + ex) \geqslant bc\psi_q(d + ex) \geqslant bc\psi_q(a + bx),$$

so

$$bc\psi_q(a + bx) - ef\psi_q(d + ex) \leqslant 0.$$

(ii) If $\psi_q(d + ex) > 0$, considering (2.3) we see that there are two possibilities for $\psi_q(a + bx)$.

Case 1. $\psi_q(a + bx) < 0$, *Case 2.* $\psi_q(a + bx) > 0$.

Hence we have:

Case 1. $bc\psi_q(a + bx) < 0$ and $ef\psi_q(d + ex) > 0$ so clearly (2.4) holds.

Case 2. The possibility $\psi_q(a + bx) > 0$ was proved in (i). □

Lemma 2.5. *Let a, b, c, d, e, f be real numbers such that $a + bx > 0, d + ex > 0, a + bx \leqslant d + ex$ and $bc \geqslant ef > 0$. Let $q \in (0, 1)$. If*

(i) $\psi_q(d + ex) < 0$ *or*

(ii) $\psi_q(a + bx) < 0$

then

$$bc\psi_q(a + bx) - ef\psi_q(d + ex) \leqslant 0. \tag{2.5}$$

Proof. (i) Let $\psi_q(d+ex) < 0$. From Lemma 2.3 we have $\psi_q(a+bx) \leqslant \psi_q(d+ex) < 0$. Multiplying both sides of inequality $bc \geqslant ef$ with $\psi_q(a + bx)$ we obtain

$$bc\psi_q(a + bx) \leqslant ef\psi_q(a + bx) \leqslant ef\psi_q(d + ex),$$

so

$$bc\psi_q(a + bx) - ef\psi_q(d + ex) \leqslant 0.$$

(ii) If $\psi_q(a + bx) < 0$, considering (2.3) we find out that there are two possibilities for $\psi_q(d + ex)$.

Case 1. $\psi_q(d + ex) > 0$, *Case 2.* $\psi_q(d + ex) < 0$.

Then we proceed in the same way as in previous lemma. □

Theorem 2.6. *Let f be a function defined by*

$$f(x) = \frac{\Gamma_q(a+bx)^c}{\Gamma_q(d+ex)^f}, \quad x \geqslant 0,\ q \in (0,1) \tag{2.6}$$

where a, b, c, d, e, f are real numbers such that: $a + bx > 0$, $d + ex > 0$, $a + bx \leqslant d + ex$, $ef \geqslant bc > 0$. If $\psi_q(a + bx) > 0$ or $\psi_q(d + ex) > 0$ then the function f is decreasing for $x \geqslant 0$. For $x \in [0, 1]$ the following double inequality holds:

$$\frac{\Gamma_q(a+b)^c}{\Gamma_q(d+e)^f} \leqslant \frac{\Gamma_q(a+bx)^c}{\Gamma_q(d+ex)^f} \leqslant \frac{\Gamma_q(a)^c}{\Gamma_q(d)^f}. \tag{2.7}$$

Proof. Let g be a function defined by $g(x) = \log f(x)$. Then

$$g(x) = c \log \Gamma_q(a+bx) - f \log \Gamma_q(d+ex).$$

So

$$g'(x) = bc\frac{\Gamma_q'(a+bx)}{\Gamma_q(a+bx)} - ef\frac{\Gamma_q'(d+ex)}{\Gamma_q(d+ex)} = bc\psi_q(a+bx) - ef\psi_q(d+ex).$$

By (2.4), we have $g'(x) \leqslant 0$. It means g is decreasing for $x \geqslant 0$, hence f is decreasing for $x \geqslant 0$. For $x \in [0, 1]$ we have $f(1) \leqslant f(x) \leqslant f(0)$ or

$$\frac{\Gamma_q(a+b)^c}{\Gamma_q(d+e)^f} \leqslant \frac{\Gamma_q(a+bx)^c}{\Gamma_q(d+ex)^f} \leqslant \frac{\Gamma_q(a)^c}{\Gamma_q(d)^f}.$$

This concludes the proof of the Theorem. □

In a similar way, using Lemma 2.5 it is easy to prove the following theorem.

Theorem 2.7. *Let f be a function defined by*

$$f(x) = \frac{\Gamma_q(a+bx)^c}{\Gamma_q(d+ex)^f} \quad x \geqslant 0, q \in (0,1) \tag{2.8}$$

where a, b, c, d, e, f are real numbers such that: $a + bx > 0$, $d + ex > 0$, $a + bx \leqslant d + ex$, $bc \geqslant ef > 0$. If $\psi_q(d + ex) < 0$ or $\psi_q(a + bx) < 0$ then the function f is decreasing for $x \geqslant 0$. For $x \in [0, 1]$ the inequality (2.7) holds.

By Theorems 2.6 and 2.7 and using (1.3) it is easy to verify that the following remarks hold:

Remark 2.8. Considering (2.7) with $a = 1$, $b = 1$, $c = n$, $n \in \mathbb{N}$, $d = 1$, $e = n$, $n \in \mathbb{N}$, $f = 1$ and (1.3) one obtains the q-analogue to the inequality given in [1], which was proved in [9]

Remark 2.9. Considering (2.7) with $a = 1$, $b = 1$, $c = a$, $a \geqslant 1$, $d = 1$, $e = a$, $f = 1$ and (1.3) one obtains the q-analogue to the inequality given in [2], also proved in [9].

Remark 2.10. If in (2.7) we take $a = 1$, $c = a$, $d = 1$, $e = a$, $f = b$, with $c \geqslant f > 0$ and using (1.3) we obtain q-analogue to the inequality given in [3].

Remark 2.11. If in (2.7) we take $a = b$, $b = a$, $c = d$, $d = a$, $e = b$, $f = c$, $ef \geqslant bc > 0$, with $a \geqslant b > 0$ and $\psi_q(b + ax) > 0$, as well as using (1.3) we obtain q-analogue to the inequality [4] proved in [10].

References

[1] Alsina, C., Tomás, M.S., A geometrical proof of a new inequality for the gamma function, *J. Ineq. Pure Appl. Math.*, 6(2) (2005) Article 48.

[2] Sándor, J., A note on certain inequalities for the gamma function, *J. Ineq. Pure Appl. Math.*, 6(3) (2005) Article 61.

[3] Bougoffa, L., Some inequalities involving the gamma function, *J. Ineq. Pure Appl. Math.*, 7(5) (2006) Article 179.

[4] Shabani, A.Sh., Some inequalities for the gamma function, *J. Ineq. Pure Appl. Math.*, 8(2) (2007) Article 49.

[5] Shabani, A.Sh., Generalization of some inequalities for the gamma function, accepted for publication in *Mathematical Communications*.

[6] Chaudhry, M.A., Zubair, S.M., A class of incomplete gamma function with applications, *CRC Press*, 2002.

[7] Grinshpan, A.Z., Ismail, M.E.H., Completely monotonic functions involving the gamma and q-gamma functions, *Proc. Amer. Math. Soc.*, 134 (2005) 1153–1160.

[8] Askey, R., The q-gamma and q-beta functions, *Applicable Anal.*, 8(2) (1978/79) 125–141.

[9] Kim, T., Adiga, C., On the q-analogue of the gamma function and related inequalities, *J. Ineq. Pure Appl. Math.*, 6(4) (2005) Article 118.

[10] Mansour, T., Some inequalities for the q-gamma function, *J. Ineq. Pure Appl. Math.*, 9(1) (2008) Article 18.

Armend Sh. Shabani
Department of Mathematics
University of Prishtina
Prishtinë 10000
Republic of Kosova
e-mail: armend_shabani@hotmail.com

Annales Mathematicae et Informaticae
***35** (2008) pp. 135–146*
http://www.ektf.hu/ami

About the geometry of milling paths

Márta Szilvási-Nagy[1][a], Szilvia Béla[a], Gyula Mátyási[b]

[a] Department of Geometry
Budapest University of Technology and Economics

[b] Department of Manufacturing Science and Technology
Budapest University of Technology and Economics

Submitted 15 June 2007; Accepted 7 May 2008

Abstract

In computer-aided manufacturing systems a number of methods have been published for milling path generation considering different geometric requirements and conditions determined by specific environments. In this paper we propose a method for the computation of the moving direction of the cutting tool in 3-axis milling considering the local features of the surface. Our method combines two geometric approaches. Computations are presented on analytical surfaces and on triangle meshes.

Keywords: 3-axis milling, tool path generation, isophotic lines, triangle mesh

MSC: 68U05, 68U07, 65D17, 65D18

1. Introduction

The most frequently used toolpath generation methods in CNC machining of free-form surfaces use planar curves, where the surface is intersected with parallel planes, and the intersection curves are taken as tool paths. The distance between two adjacent intersecting planes determines the distance between two tool paths (called tool path side step). Between the tool paths a scallop (rib) arises, the height of which measures the machining error (Figures 1, 2). This error depends on the shape of the cutting tool (flat end, ball end or other shapes) and also on the shape of the surface. Several computations have been published for optimizing the total length of the tool paths which is longer, if the number of cutting planes are larger, while the error is within a prescribed tolerance.

[1]Supported by a joint project between the TU Berlin and the BUTE and by the Hungarian National Foundation OTKA No. T047276

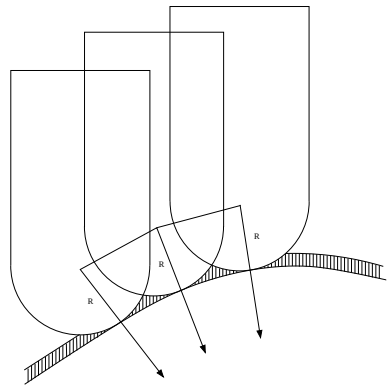

Figure 1: Machining error arising with ball end tool

The change of the angle between the tool axis and the surface normal leads to unevenly distributed side steps and fluctuating errors. A solution of this problem is the segmentation of the surface into regions, where this angle does not change too much. Such a segmentation with isophotic lines is proposed in [2]. Then the distances of the intersecting planes are computed in each region according to the prescribed tolerance (Figure 3). Within such a region the contact surface of the tool end with the material to be removed has a constant size. Consequently, the abrasion of the tool is even. An isophote interpolation method is proposed in [10], which can be used in isophote based tool path generations.

In an other strategy the tools are moving on the surface along isoparametric curves instead of plane sections [4]. Based on curvature values, independently from the parametrization of the surface, local and global millability with a given tool are investigated in [8] and [9]. This strategy takes into account that the width of the machined stripe depends on the curvature of the surface, and proposes a moving direction of the tool in which the stripe is the widest.

Fig 4 shows which part of the material is removed by the milling tool. The common part of the offset surface determined by the prescribed machining tolerance ε and the ball end of the tool is the so called contact surface. The projection of its boundary curve onto the surface determines the width s of the machining stripe, which is wider on flatter surfaces and narrower on more curved surfaces.

In this paper we propose a method for tool path generation considering the following requirements: (i) the change of the angle between the surface normal and the tool axis along a tool path is minimal, and (ii) the side step length is maximal, while the machining error is smaller than a given tolerance. Of course, these requirements cannot be fulfilled at the same time, we try to find a compromise.

Our investigations are restricted to 3-axis milling with ball end tools. We present our computations on analytical and on discrete surfaces.

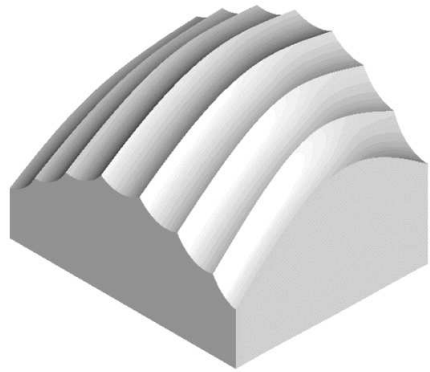

Figure 2: Processed surface with scallops

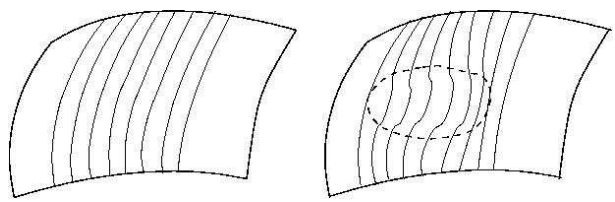

Figure 3: Segmentation of a surface with varying curvature

2. Analytic surfaces

We assume that the analytic surface is represented by a function $f(x, y)$ over a region in the xy plane, and the axis of the cutter is parallel to the z-axis. Our task is to determine the moving direction of the tool from every point of the surface considering the requirements of tool path generation and the geometrical features of the surface. In our investigations two geometric requirements will influence the shape of milling paths.

The first geometrical requirement is to ensure even abrasion of the tool end. This means that we want to keep the angle between the tool axis and surface normal constant during the cutting motion. A curve on the surface in the points of which the surface normal and a reference direction (here the tool axis) form a constant angle is called isophote or isophotic curve. On a smooth surface an isophotic curve assigned to an angle between zero and 90 degrees is a continuous curve, or a point. On the other hand, each point of the surface belongs to an isophotic curve, or the surface normal is parallel to the reference direction at this point. Properties and computation methods of isophota are described in [5]. In Fig 5 isophotic curves are shown on a quadratic surface by sequences of points which are computed on

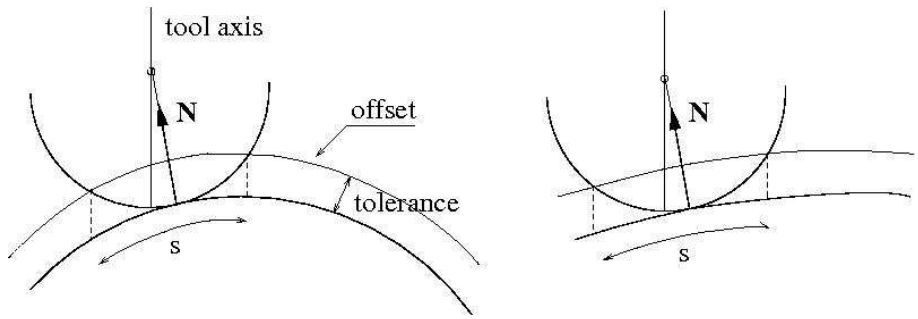

Figure 4: Simplified axial intersection of the milling tool and the surface

the base of the definition with a given step size.

The distances between adjacent isophota on a generic surface are varying, therefore the fluctuation of the machining error will be out of control, when taking isophota for tool paths.

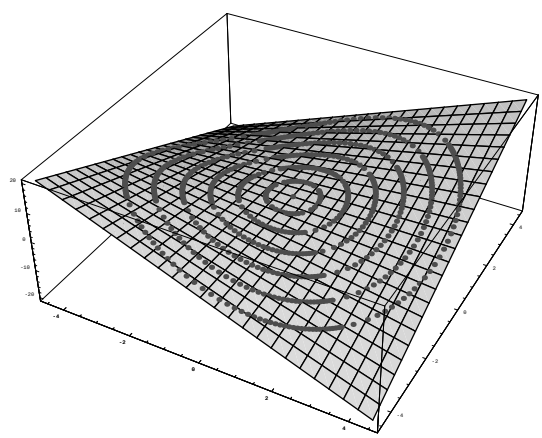

Figure 5: Isophota on the saddle surface

We have to keep the machining error within a given tolerance. Therefore, we first compute the processed part of the surface around a contact point of the ball end and the surface. The error is less than a given tolerance ε, if the points of the processed part are between the task surface $f(x, y)$ and its offset $\mathit{Off}\,(f, \varepsilon, x, y)$ consisting of the points which have the distance ε to the surface $f(x, y)$. If the coordinates of the contact point are $(x_0, y_0, f(x_0, y_0))$, then we obtain the equation

$$|\mathit{Off}\,(f, \varepsilon, x, y) - \mathit{Off}\,(f, R, x_0, y_0)| = R \tag{2.1}$$

for the boundary curve (called contact curve) of the processed part which is the curve of intersection of the ball end and the offset surface, where $\mathit{Off}\,(f, \varepsilon, x, y)$ is the offset of the surface $f(x, y)$ with distance ε, and the second term gives the center of the ball end. The points of the contact curve are determined by the solutions (x, y) of this equation. (Our numerical method will result in a given number of points.) The normal projection of the contact curve on the surface determines the surface patch of our interest. This is the processed surface patch, and the optimal moving direction of the tool at the actual contact point is perpendicular to the largest diameter of this patch. In this way we get the widest machined stripe.

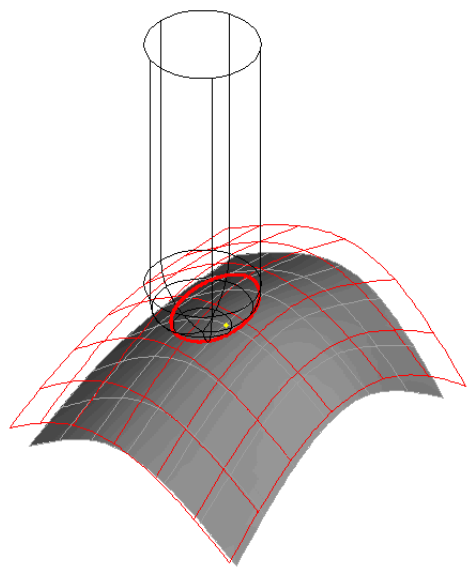

Figure 6: The boundary curve on the offset surface of the processed patch

We can calculate this direction in different ways. One method given in [9] calculates with the difference of the surface of the cutting tool and the task surface $f(x, y)$. This difference surface is approximated in second order, then the boundary curve of the processed patch, for the points of which the approximated difference is less than the tolerance, is projected onto the xy plane. The obtained curve is an ellipse, and its major axis determines the largest width of the machined stripe, while its minor axis determines the proposed moving direction.

In our geometric approach we approximate the moving direction from the equation (2.1). We estimate the diameters of the processed patch in n different directions (n is a given number in our algorithm). In this calculation we use in the above equation (2.1) the Taylor polynomial of degree 8 of the surface $f(x, y)$. First, we

set up n directions around the contact point and a vertical plane through each direction at the contact point. Then for each plane we solve the system of equations formed by (2.1) and the actual plane numerically. The solution gives the (x, y) coordinates of the end points of the patch diameter in this plane. Finally, we choose the direction of the largest diameter, and the proposed moving direction is perpendicular to it.

Now we want to consider the two requirements at the same time. That is, the tool should process a wide stripe, while the abrasion of the tool is even. Our compromise is the following. We modify the moving direction computed from the first requirement in the following way. We compute the isophote passing through the actual contact point. Then we move the tool end neither along this isophote, nor in the computed moving direction, but along a bisector direction of them. According to the two possible orientations of the isophote two bisectors exist. One is in "forward direction", the other one "backwards". The isophote passing through the actual contact point intersects the boundary curve of the processed surface patch in two points (Fig 7). The appropriate direction can be selected with the help of the two points of intersection and the moving direction computed in the former step. We have chosen the next contact point in this corrected moving direction by a specified constant step length.

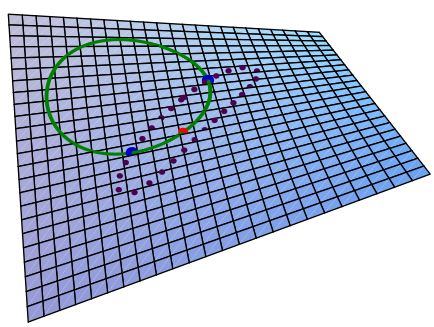

Figure 7: Isophote (the left side curve) and the boundary of the processed patch at the contact point

In our further investigations we'll try to compute the correction of the moving direction by the isophotic curves and also the step distance considering the local shape of the surface. The overlapping of the processed patches along the adjacent tool paths require further investigations too.

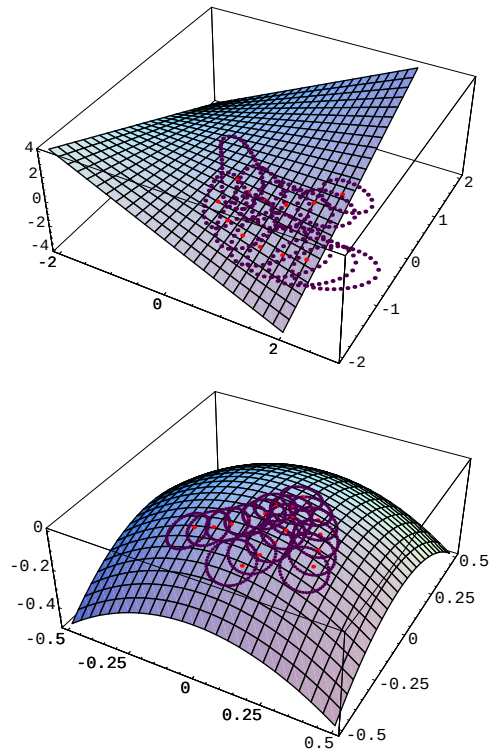

Figure 8: Processed patches along tool paths on the saddle surface and on the sphere

3. Discrete surfaces

Computation of characteristic values of a free-form surface approximated by a triangular mesh requires quite a different technique from that in the analytic case. Namely, the surface data can be only estimated from the mesh data. Standard representations of triangle meshes are generated by the most CAD systems in STL (stereo lithography) format developed for rapid prototyping. Such an STL data structure contains the set of the mesh triangles, which complemented with adjacency informations becomes a polyhedral data structure. A polyhedral data structure makes possible to compute the line of intersection of the mesh with a plane which is the base of different algorithms, e.g. slicing in layered manufacturing [6] or tool path generation in milling. Though discrete counterparts of differential operators have been developed and published, there are no unique or best methods for estimating the surface normal or the curvature values at the points of the mesh.

For the characterization of the local shape of a surface presented by a triangle

mesh the estimation of principal directions is crucial. We apply in our computations the method of geodesic disk described in [7] (Fig 9). In this method the normal curvature values are estimated at the center point of a given triangle of the mesh in the following way. The mesh is intersected by a set of normal planes passing through the barycentric center of the triangle, and in each normal plane a fixed geodesic radius is measured along the polygonal line of intersection in both directions from the center point. The chord length of such a geodesic diameter characterizes the normal curvature in this intersecting plane. The normal curvature approximated from the geodesic radius r_g and the chord length d is (Fig 10)

$$\kappa_n \approx \frac{1}{r_g}\sqrt{6\left(1-\frac{d}{2r_g}\right)}. \tag{3.1}$$

Selecting the maximal normal curvature, i.e. the minimal chord length at the given point determines one of the two principal directions. In Fig 11 a geodesic circle is shown on the mesh of the duck. On the right hand side only feature and silhouette edges are drawn. The principal direction of the biggest normal curvature is indicated by a straight line segment.

We note that the method of the geodesic circle is suitable also for detecting planar and spherical regions on the mesh.

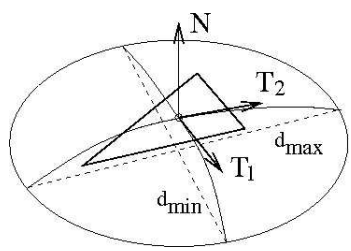

Figure 9: Geodesic disk for estimating normal curvatures and principal directions

In generating the offset of a triangle mesh several problems arise. After moving the facets in their normal directions by a given distance, gaps and overlappings occur in convex and concave regions, respectively. One method for avoiding gaps is offsetting also edges and vertices in averaged normal vector directions, then trimming the adjacent surfaces to each other [3]. Trimming and removing the overlapping portions are made in complicated processes. In an other approach, where tool paths are generated in parallel driving planes, filling of gaps is made in plane sections of the offsetted facets with the driving planes, then arc and line segments are used. The trimming problem is solved also in two dimensions in order to generate a smooth tool path in the actual plane [1].

In our method we solve the offsetting problem in normal sections. In order to determine the processed patch with a ball end on the mesh, the contact curve, i.e.

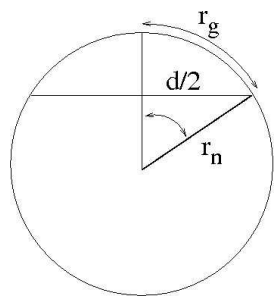

Figure 10: Normal curvature estimated in a normal section

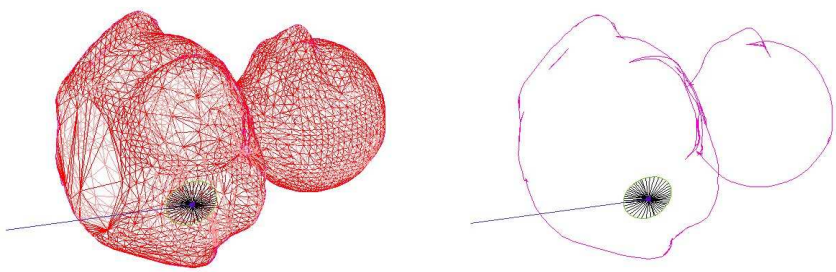

Figure 11: Triangle mesh of the duck and a principle direction

the curve of intersection of the surface of the ball end with the offset mesh has to be computed, then this curve has to be projected onto the mesh. We compute the points of the contact curve in a set of normal planes in the following way. 1. We set up n normal planes through the contact point. 2. We intersect the triangle face of the contact point and its two neighbours with the actual normal plane. 3. We move the obtained segments in the normal direction of the intersected triangles by the distance of the prescribed tolerance. 4. We fill the gap between the offset segments, or we remove the overlapping parts (Fig 12). 5. Along the polygonal line obtained in this way we measure the distance of the moving point from the ball end center. If this distance is equal to the ball end radius, then the point is on the contact curve. If all such distances are smaller than the radius, we intersect the neighbouring triangles with the normal plane in both directions, and we repeat the last three steps. 6. We project the two points of the contact curve computed in the actual normal plane onto the mesh. Finally, we get $2n$ boundary points of the processed patch around the contact point. We remark that our local offsetting method works with two dimensional algorithms.

The result of this computation is shown on a "real" triangle mesh of a sphere. The floating surface patch shown with 24 diameters is the part of the offset mesh

intersected with the ball end (Fig 13). Its projection on the mesh is the processed patch with the given tolerance. The perpendicular direction to the direction of the largest diameter of this patch gives a moving direction in which the widest machined stripe arises. This moving direction is to be corrected by minimizing the change of the surface normal direction within a prescribed angular neighborhood, if also the requirement of even abrasion of the tool is considered.

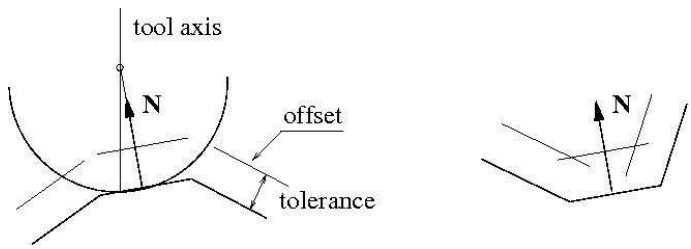

Figure 12: Offsetting in a normal plane

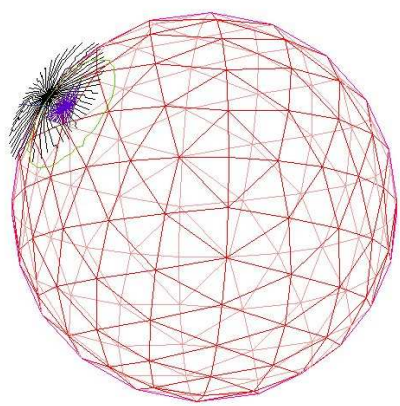

Figure 13: Intersection of a ball end with the offset of the sphere

4. Conclusions

In this paper a method is represented for the computation of the moving direction of a ball end tool in 3-axis milling. In this method two geometric requirements are considered at the same time, and a compromizing solution is proposed to meet both of them.

The computations and the figures are made with the algebraic symbolic program package Mathematica in the case of analytical description of the surface. The algorithms are implemented in the program language Java in the case of triangle meshes.

References

[1] Chuang, C.-M., Yau, H.-T., A new approach to z-level contour machining of triangulated surface models using fillet endmills, *Computer-Aided Design*, Vol. 37 (2005) 1039–1051.

[2] Ding, S., Mannan, M.A., Poo, A.N., Yang, D.C.H., Han, Z., Adaptive isoplanar tool path generation for machining of free-form surfaces, *Computer-Aided Design*, Vol. 35 (2003) 141–153.

[3] Jun, C.S., Kim, D.S., Park, S., A new curve-based approach to polyhedral machining, *Compuer-Aided Design*, Vol. 34 (2002) 379–389.

[4] Elber, G., Cohen, E., Tool path generation for free form surface models, *Compuer-Aided Design*, Vol. 26 (1994) 490–496.

[5] Lang, J., Zur Konstruktion von Isophoten im Computer Aided Design, *CAD-Computergraphik und Konstruktion*, Vol. 33., Wien 1984.

[6] Szilvasi-Nagy, M., Removing errors from triangle meshes by slicing, *Third Hungarian Conference on Computer Graphics and Geometry*, (17–18. Nov. 2005 Budapest, Hungary) 125–127.

[7] Szilvasi-Nagy, M., About curvatures on triangle meshes, *KoG*, 10 (2007) 13-18.

[8] Glaeser, G., Wallner, J., Pottmann, H., Collision-Free 3-Axis Milling and Selection of Cutting-Tools, *Computer-Aided Design*, Vol. 31 (1999) 225–232.

[9] Wallner, J., Glaeser, G., Pottmann, H., Geometric contributions to 3-axis milling of sculptured surfaces, In Machining Impossible Shapes (G. Olling, B. Choi and R. Jerard, eds.), pp. 33–41, Boston: Kluwer Academic Publ., (1999)

[10] Xu, H.Y., Tam, H.Y., Zang, J.J., Isophote interpolation, *Compuer-Aided Design*, Vol. 35 (2003) 1337–1344.

Márta Szilvási-Nagy
Szilvia Béla
Department of Geometry
Budapest University of Technology and Economics
H-1111 Budapest
Egry József u. 1. H. 22.

e-mail:
szilvasi@math.bme.hu
belus@math.bme.hu

Gyula Mátyási
Department of Manufacturing Engineering

Budapest University of Technology and Economics
H-1111 Budapest
Egry József u. 1. E. 2. 11.

e-mail:
matyasi@manuf.bme.hu

Annales Mathematicae et Informaticae
***35** (2008) pp. 147–154*
http://www.ektf.hu/ami

Convergence rate in the strong law of large numbers for mixingales and superadditive structures

Tibor Tómács

Department of Applied Mathematics
Eszterházy Károly College, Eger, Hungary

Submitted 15 September 2008; Accepted 30 October 2008

Abstract

In this paper we study convergence rates in the strong laws of large numbers for mixingales and superadditive structures, by using a general method.

Keywords: convergence rate, strong law of large numbers, L^r mixingale, sequence with superadditive moment function

MSC: 60F15

1. Introduction

Sung, Hu and Volodin [8] introduced a new method for obtaining convergence rate in the strong law of large numbers (SLLN), by using the approach of Fazekas and Klesov [2]. This result generalizes and sharpens the method of Hu and Hu [5]. Tómács [9] gave a general method by using a Hájek–Rényi type inequality (see Hájek and Rényi [3]) for the probabilities, which sharpens the result of Sung, Hu and Volodin [8]. In this paper we apply this method for mixingales and superadditive structures.

The concept of L^2 mixingales was introduced by McLeish [6], and generalized to L^r mixingales by Andrews [1]. The definition of superadditive moment function is due to Móricz [7].

Fazekas and Klesov [2, Theorem 6.1 and 6.2] proved SLLN's for mixingales. In Section 3 we shall give the convergence rates in these SLLN's. Hu and Hu [5, Theorem 2.1] obtained convergence rate in SLLN under the superadditivity property. In Section 4 we shall generalize this result.

We use the following notation. Let $\mathbb{N}$ be the set of the positive integers and $\mathbb{R}$ the set of 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

$\sum_{k\in A} a_k = 0$. In this paper let $\{X_k,\ k \in \mathbb{N}\}$ be a sequence of random variables defined on a fixed probability space $(\Omega, \mathcal{F}, \mathrm{P})$, $S_n = \sum_{k=1}^{n} X_k$ for all $n \in \mathbb{N}$ and $S_0 = 0$. Finally in this paper let $\{b_k,\ k \in \mathbb{N}\}$ be a nondecreasing unbounded sequence of positive real numbers.

2. A general method to obtain the rate of convergence in the SLLN

Definition 2.1. Let Θ_r $(r > 0)$ denote the set of functions $\vartheta\colon [0,\infty) \to \mathbb{R}$ which are nondecreasing, continuous at 0, $\vartheta(0) = 0$, $\vartheta(x) > 0$ for all $x > 0$ and

$$\sum_{n=1}^{\infty} n^{-2}\vartheta^{-r}(n^{-1}) < \infty.$$

Remark 2.2. It is easy to see that if $0 < \delta < 1$ and $\vartheta(x) = x^{\delta/r}$ $(x \geqslant 0)$, then $\vartheta \in \Theta_r$.

Theorem 2.3 (Tómács [9], Theorem 3.4). *Let $\{\alpha_k,\ k \in \mathbb{N}\}$ be a sequence of nonnegative real numbers, $r > 0$ and*

$$\beta_n = \max_{k\leqslant n} b_k \vartheta\left(\sum_{i=k}^{\infty} \alpha_i b_i^{-r}\right), \quad \text{where} \quad \vartheta \in \Theta_r. \tag{2.1}$$

If

$$\sum_{k=1}^{\infty} \alpha_k b_k^{-r} < \infty \tag{2.2}$$

and there exists $c > 0$ such that for any $n \in \mathbb{N}$ and any $\varepsilon > 0$

$$\mathrm{P}\Big(\max_{k\leqslant n} |S_k| \geqslant \varepsilon\Big) \leqslant c\varepsilon^{-r} \sum_{k=1}^{n} \alpha_k, \tag{2.3}$$

then

$$\lim_{n\to\infty} \frac{\beta_n}{b_n} = 0 \quad \text{and} \quad \frac{S_n}{b_n} = O\left(\frac{\beta_n}{b_n}\right) \quad \text{almost surely (a.s.).}$$

Lemma 2.4. *Let $\{\alpha_k,\ k \in \mathbb{N}\}$ be a sequence of nonnegative real numbers, $r > 0$, $0 < \delta < 1$, $\vartheta(x) = x^{\delta/r}$ for all $x \geqslant 0$, $b_k = k^{1/r}$ for all $k \in \mathbb{N}$ and let β_n be defined by (2.1). If there exist $c > 0$ and $0 < \gamma < 1$ such that $\sum_{i=k}^{\infty} \alpha_i/i \leqslant c\sum_{i=k}^{\infty} i^{-1-\gamma}$ for all $k \in \mathbb{N}$, then*

$$\frac{\beta_n}{n^{1/r}} = O\left(\frac{1}{n^{\gamma\delta/r}}\right).$$

Proof. Since $\sum_{i=k}^{\infty} i^{-1-\gamma} \leqslant \int_{k-1}^{\infty} x^{-1-\gamma}\,\mathrm{d}x = \gamma^{-1}(k-1)^{-\gamma}$ for all $k \geqslant 2$, hence we get

$$\sum_{i=k}^{\infty} \frac{\alpha_i}{i} \leqslant \frac{c}{\gamma(k-1)^{\gamma}} \quad \text{for all} \quad k \geqslant 2 \tag{2.4}$$

and

$$\sum_{i=1}^{\infty} \frac{\alpha_i}{i} \leqslant c \sum_{i=1}^{\infty} i^{-1-\gamma} = c + c \sum_{i=2}^{\infty} i^{-1-\gamma} \leqslant c + \frac{c}{\gamma(2-1)^{\gamma}} = \frac{c}{\gamma}(\gamma+1).$$

It follows that

$$\beta_1 = \left(\sum_{i=1}^{\infty} \frac{\alpha_i}{i} \right)^{\delta/r} \leqslant \left(\frac{c}{\gamma}(\gamma+1) \right)^{\delta/r} \leqslant \left(\frac{c2^{\gamma}}{\gamma}(\gamma+1) \right)^{\delta/r}. \tag{2.5}$$

On the other hand if $n \geqslant 2$ then (2.4) implies

$$\max_{2 \leqslant k \leqslant n} k^{1/r} \left(\sum_{i=k}^{\infty} \frac{\alpha_i}{i} \right)^{\delta/r} \leqslant \max_{2 \leqslant k \leqslant n} k^{1/r} \left(\frac{c}{\gamma(k-1)^{\gamma}} \right)^{\delta/r}$$
$$\leqslant \max_{2 \leqslant k \leqslant n} \left(\frac{c2^{\gamma}}{\gamma} \right)^{\delta/r} k^{(1-\gamma\delta)/r} = \left(\frac{c2^{\gamma}}{\gamma} \right)^{\delta/r} n^{(1-\gamma\delta)/r}.$$

This inequality, (2.5) and $\lim_{n\to\infty} n^{(1-\gamma\delta)/r} = \infty$ imply for $n \in \mathbb{N}$ large enough

$$\beta_n \leqslant \text{const.} \max \left\{ (\gamma+1)^{\delta/r}, n^{(1-\gamma\delta)/r} \right\} = \text{const.} n^{(1-\gamma\delta)/r}.$$

So $\beta_n n^{-1/r} \leqslant \text{const.} n^{-\gamma\delta/r}$ for $n \in \mathbb{N}$ large enough, which implies the statement. □

3. Mixingales

Let $\{\mathcal{F}_k,\ k \in \mathbb{N}\}$ be a nondecreasing sequence of sub σ-fields of $\mathcal{F}$, $\mathrm{E}_m X_k = \mathrm{E}(X_k \mid \mathcal{F}_m)$ denote the conditional expectation of X_k given $\mathcal{F}_m$ for $m > 0$ and $\mathrm{E}_m X_k = 0$ for $m \leqslant 0$.

Definition 3.1 (McLeish [6], Andrews [1]). The sequence $\{(X_k, \mathcal{F}_k), k \in \mathbb{N}\}$ is an L^r *mixingale* if there exist nonnegative constants $\{c_k, k \geqslant 0\}$ and $\{\psi_k, k \geqslant 0\}$ such that $\psi_k \downarrow 0$ and for all nonnegative integers k and m we have

$$||\,\mathrm{E}_{k-m} X_k||_r \leqslant c_k \psi_m \quad \text{and} \quad ||X_k - \mathrm{E}_{k+m} X_k||_r \leqslant c_k \psi_{m+1},$$

where $||\xi||_r = (\mathrm{E}\,|\xi|^r)^{1/r}$ for any random variable ξ.

Lemma 3.2. *If* $\{(X_k, \mathcal{F}_k), k \in \mathbb{N}\}$ *is* L^r mixingale*, where* $r \geqslant 2$ *and* $\sum_{m=1}^{\infty} \psi_m < \infty$*, then there exists* $c > 0$ *such that for any* $n \in \mathbb{N}$ *and any* $\varepsilon > 0$

$$\mathrm{P}\Big(\max_{k \leqslant n} |S_k| \geqslant \varepsilon\Big) \leqslant c\varepsilon^{-r} \left(\sum_{k=1}^{n} c_k^2\right)^{r/2}.$$

Proof. Hansen [4] proved in Lemma 2 under these conditions, that there exists $c > 0$ such that for any $n \in \mathbb{N}$

$$\mathrm{E}\Big(\max_{k \leqslant n} |S_k|^r\Big) \leqslant c \left(\sum_{k=1}^{n} c_k^2\right)^{r/2}.$$

Hence Markov's inequality implies the statement. □

Theorem 3.3. *Let* $\{(X_k, \mathcal{F}_k), k \in \mathbb{N}\}$ *be an* L^r mixingale*, where* $r \geqslant 2$ *and* $\sum_{m=1}^{\infty} \psi_m < \infty$*. Let* β_n *defined by* (2.1) *with*

$$\alpha_k = \left(\sum_{i=1}^{k} c_i^2\right)^{r/2} - \left(\sum_{i=1}^{k-1} c_i^2\right)^{r/2}.$$

If

$$\sum_{k=1}^{\infty} \frac{c_k^2}{b_k^r} \left(\sum_{i=1}^{k} c_i^2\right)^{r/2-1} < \infty, \tag{3.1}$$

then

$$\lim_{n\to\infty} \frac{\beta_n}{b_n} = 0 \quad \text{and} \quad \frac{S_n}{b_n} = O\left(\frac{\beta_n}{b_n}\right) \quad \text{a.s.}$$

Proof. If $A = \emptyset$ then $\sum_{i \in A} c_i^2 = 0$, hence $\alpha_1 = c_1^r$. Since

$$\sum_{k=1}^{n} \alpha_k = \left(\sum_{i=1}^{n} c_i^2\right)^{r/2},$$

hence Lemma 3.2 implies (2.3). By the mean value theorem

$$x_2^{r/2} - x_1^{r/2} \leqslant (x_2 - x_1)\frac{r}{2} x_2^{r/2-1} \quad \text{for all} \quad 0 \leqslant x_1 \leqslant x_2. \tag{3.2}$$

Using (3.2) with $x_1 = \sum_{i=1}^{k-1} c_i^2$ and $x_2 = \sum_{i=1}^{k} c_i^2$ we get

$$\alpha_k = x_2^{r/2} - x_1^{r/2} \leqslant c_k^2 \frac{r}{2} \left(\sum_{i=1}^{k} c_i^2\right)^{r/2-1}.$$

This inequality and (3.1) imply (2.2). Since every conditions of Theorem 2.3 are satisfied, the statement is proved. □

Lemma 3.4. *If* $\{(X_k, \mathcal{F}_k), k \in \mathbb{N}\}$ *is an* L^r mixingale, *where* $1 < r \leqslant 2$ *and* $\sum_{m=1}^{\infty} \psi_m < \infty$, *then there exists* $c > 0$ *such that for any* $n \in \mathbb{N}$ *and any* $\varepsilon > 0$

$$\mathrm{P}\Big(\max_{k \leqslant n} |S_k| \geqslant \varepsilon\Big) \leqslant c\varepsilon^{-r} \sum_{k=1}^{n} c_k^r.$$

Proof. Hansen [4] proved in Lemma 2 of Erratum under these conditions, that there exists $c > 0$ such that for any $n \in \mathbb{N}$

$$\mathrm{E}\Big(\max_{k \leqslant n} |S_k|^r\Big) \leqslant c \sum_{k=1}^{n} c_k^r.$$

Hence Markov's inequality implies the statement. □

Theorem 3.5. *Let* $\{(X_k, \mathcal{F}_k), k \in \mathbb{N}\}$ *be an* L^r mixingale, *where* $1 < r \leqslant 2$ *and* $\sum_{m=1}^{\infty} \psi_m < \infty$. *Let* β_n *defined by* (2.1) *with* $\alpha_k = c_k^r$. *If*

$$\sum_{k=1}^{\infty} \frac{c_k^r}{b_k^r} < \infty, \tag{3.3}$$

then

$$\lim_{n \to \infty} \frac{\beta_n}{b_n} = 0 \quad \text{and} \quad \frac{S_n}{b_n} = O\left(\frac{\beta_n}{b_n}\right) \quad \text{a.s.}$$

Proof. The statement is a corollary of Lemma 3.4 and Theorem 2.3. □

Corollary 3.6. *Let* $\{(X_k, \mathcal{F}_k), k \in \mathbb{N}\}$ *be an* L^r mixingale, *where* $1 < r \leqslant 2$ *and* $\sum_{m=1}^{\infty} \psi_m < \infty$. *If there exist* $c > 0$ *and* $0 < \gamma < 1$ *such that* $c_k \leqslant ck^{-\gamma/r}$ *for all* $k \in \mathbb{N}$, *then for all* $0 < \delta < 1$

$$\frac{S_n}{n^{1/r}} = O\left(\frac{1}{n^{\gamma\delta/r}}\right) \quad \text{a.s.}$$

Proof. Let $b_k = k^{1/r}$, $\alpha_k = c_k^r$ and $\vartheta(x) = x^{\delta/r}$ $(x \geqslant 0)$, where $0 < \delta < 1$ is a fixed constant. Then for all $k \in \mathbb{N}$

$$\sum_{i=k}^{\infty} \frac{\alpha_i}{i} = \sum_{i=k}^{\infty} \left(\frac{c_i}{b_i}\right)^r \leqslant c^r \sum_{i=k}^{\infty} i^{-1-\gamma}.$$

Hence using Theorem 3.5 and Lemma 2.4 we get

$$\frac{S_n}{n^{1/r}} = O\left(\frac{\beta_n}{n^{1/r}}\right) = O\left(\frac{1}{n^{\gamma\delta/r}}\right) \quad \text{a.s.}$$

□

4. Sequences with superadditive moment function

Definition 4.1 (Móricz [7]). $\{X_k,\ k \in \mathbb{N}\}$ is said to have the r-th ($r > 0$) *moment function of superadditive structure* if there exists $g\colon \mathbb{N}\cup\{0\} \times \mathbb{N} \to [0,\infty)$ such that

$$g(b,k) + g(b+k,l) \leqslant g(b,k+l) \quad \text{for all} \quad b \in \mathbb{N}\cup\{0\}, k \in \mathbb{N}, l \in \mathbb{N} \tag{4.1}$$

and for some $\alpha > 1$

$$\mathrm{E}\,|S_{b+n} - S_b| \leqslant g^\alpha(b,n) \quad \text{for all} \quad b \in \mathbb{N}\cup\{0\}, n \in \mathbb{N}. \tag{4.2}$$

We shall use the notation $g_n = g(0,n)$ ($n \in \mathbb{N}$) and $g_0 = 0$. It is easy to see that $g_n \leqslant g_{n+1}$ for all $n \in \mathbb{N}\cup\{0\}$.

Lemma 4.2. *If $\{X_k,\ k \in \mathbb{N}\}$ has r-th moment function of superadditive structure with $r > 0$, $\alpha > 1$, then there exists a constant $A_{r,\alpha}$ depending only on r and α such that for any $n \in \mathbb{N}$ and any $\varepsilon > 0$*

$$\mathrm{P}\Big(\max_{k\leqslant n} |S_k| \geqslant \varepsilon\Big) \leqslant A_{r,\alpha}\varepsilon^{-r} g_n^\alpha.$$

Proof. Móricz proved in [7] under these conditions, that there exists a constant $A_{r,\alpha}$ depending only on r and α, such that for any $n \in \mathbb{N}$

$$\mathrm{E}\Big(\max_{k\leqslant n} |S_k|^r\Big) \leqslant A_{r,\alpha} g_n^\alpha.$$

Hence Markov's inequality implies the statement. □

Theorem 4.3. *Assume that $\{X_k,\ k \in \mathbb{N}\}$ has r-th moment function of superadditive structure with $r > 0$, $\alpha > 1$. Let β_n defined by* (2.1) *with $\alpha_k = g_k^\alpha - g_{k-1}^\alpha$. If*

$$\sum_{k=1}^{\infty} \frac{g_k^\alpha - g_{k-1}^\alpha}{b_k^r} < \infty, \tag{4.3}$$

then

$$\lim_{n\to\infty} \frac{\beta_n}{b_n} = 0 \quad \text{and} \quad \frac{S_n}{b_n} = O\left(\frac{\beta_n}{b_n}\right) \quad \text{a.s.}$$

Proof. As g_k increases, we get $\alpha_k \geqslant 0$, thereby (4.3) implies (2.2). On the other hand $\sum_{k=1}^n \alpha_k = g_n^\alpha$, so Lemma 4.2 implies (2.3). Now applying Theorem 2.3 we get the statement. □

Remark 4.4. Hu and Hu proved Theorem 4.3 in special case $\vartheta(x) = x^{\delta/r}$ ($0 < \delta < 1$). (See Theorem 2.1 of Hu and Hu [5].)

Corollary 4.5. *Let $0 < \gamma < 1$, $c > 0$, $\alpha > 1$ and $r > 0$. If for all $b \in \mathbb{N} \cup \{0\}, n \in \mathbb{N}$*

$$\mathrm{E}\,|S_{b+n} - S_b| \leqslant c\left((b+n)^{(1-\gamma)/\alpha} - b^{(1-\gamma)/\alpha}\right)^{\alpha},$$

then

$$\frac{S_n}{n^{1/r}} = O\left(\frac{1}{n^{\gamma\delta/r}}\right) \quad \text{a.s. for all } \ 0 < \delta < 1.$$

Proof. Let $g\colon \mathbb{N}\cup\{0\}\times\mathbb{N}\cup\{0\} \to [0,\infty)$, $g(i,j) = c^{1/\alpha}\left((i+j)^{(1-\gamma)/\alpha} - i^{(1-\gamma)/\alpha}\right)$. Then (4.1) and (4.2) are satisfied, hence $\{X_k,\ k \in \mathbb{N}\}$ has r-th moment function of superadditive structure.

Now let $b_k = k^{1/r}$ for all $k \in \mathbb{N}$. Since $g_i^{\alpha} = g^{\alpha}(0,i) = ci^{1-\gamma}$ for every nonnegative integer i, hence we get

$$\begin{aligned}\sum_{i=k}^{\infty} \frac{g_i^{\alpha} - g_{i-1}^{\alpha}}{b_i^r} &= \sum_{i=k}^{\infty} \frac{ci^{1-\gamma} - c(i-1)^{1-\gamma}}{i} \\ &= c\sum_{i=k}^{\infty} i^{1-\gamma}\left(\frac{1}{i} - \frac{1}{i+1}\right) - c\frac{(k-1)^{1-\gamma}}{k} \leqslant c\sum_{i=k}^{\infty} i^{-1-\gamma}. \end{aligned} \tag{4.4}$$

Since (4.4) implies (4.3), hence using Theorem 4.3 we have

$$\frac{S_n}{n^{1/r}} = O\left(\frac{\beta_n}{n^{1/r}}\right) \quad \text{a.s.} \tag{4.5}$$

Let $\vartheta(x) = x^{\delta/r}$, where $0 < \delta < 1$ is a fixed constant. Then (4.4) and Lemma 2.4 imply $\beta_n/n^{1/r} = O\left(1/n^{\gamma\delta/r}\right)$. Hence we get the statement by (4.5). □

Corollary 4.6. *Let $r > 0$, $c > 0$ and $1 < \alpha < 2$. If for all $b \in \mathbb{N} \cup \{0\}$, $n \in \mathbb{N}$*

$$\mathrm{E}\,|S_{b+n} - S_b| \leqslant c\left(\sqrt{b+n} - \sqrt{b}\right)^{\alpha},$$

then

$$\frac{S_n}{n^{1/r}} = O\left(n^{-(1-\frac{\alpha}{2})\delta/r}\right) \quad \text{a.s. for all } \ 0 < \delta < 1.$$

Proof. Apply Corollary 4.5 with $\gamma = 1 - \frac{\alpha}{2}$. □

References

[1] Andrews, D.W.K., Laws of large numbers for dependent nonidentically distributed random variables, *Econometric Theory*, 4 (1988) 458–467.

[2] Fazekas, I., Klesov, O., A general approach to the strong laws of large numbers, *Theory of Probab. Appl.*, 45/3 (2000) 568–583.

[3] Hájek, J., Rényi, A., Generalization of an inequality of Kolmogorov, *Acta Math. Acad. Sci. Hungar.*, 6 no. 3–4 (1955) 281–283.

[4] Hansen, B.E., Strong laws for dependent heterogeneous processes, *Econometric Theory*, 7 (1991) 213–221; Erratum, *Econometric Theory*, 8 (1992) 421–422.

[5] Hu, S., Hu, M., A general approach rate to the strong law of large numbers, *Stat. & Prob. Letters*, 76 (2006) 843–851.

[6] McLeish, D.L., A maximal inequality and dependent strong laws, *Annals of Probability*, 3 (1975) 829–839.

[7] Móricz, F., Moment inequalities and the strong of large numbers, *Z. Wahrscheinlichkeitstheorie verw. Gebiete*, 35 (1976) 299-314.

[8] Sung, S.H., Hu, T.-C., Volodin, A., A note on the growth rate in the Fazekas-Klesov general law of large numbers and on the weak law of large numbers for tail series, *Publicationes Mathematicae Debrecen*, 73/1-2 (2008) 1–10.

[9] Tómács, T., A general method to obtain the rate of convergence in the strong law of large numbers, *Annales Mathematicae et Informaticae*, 34 (2007) 97–102.

Tibor Tómács
Department of Applied Mathematics
Eszterházy Károly College
P.O. Box 43
H-3301 Eger
Hungary

e-mail: `tomacs@ektf.hu`

Methodological papers

Annales Mathematicae et Informaticae
***35** (2008) pp. 157–162*
http://www.ektf.hu/ami

Modelling a simple continuous-time system

Gábor Geda

Department of Computer Science
Eszterházy Károly College, Eger, Hungary

Submitted 15 September 2008; Accepted 8 December 2008

Abstract

The aim of the present paper is to give a very simple example how we can set up a mathematical model describing a not too complicated phenomenon based on measurement. It may help the beginners to model other systems too, by differential equations. At the some time we would like to enrich the possibility of demonstration in this field.

Keywords: Differential equation, mathematical model, crystal growth.

MSC: 31A35, 34A30, 03H10, 97D10, 97D40, 92F05.

1. Introduction

The authors of researches dealing with studying differential equations, composing them and differential equational models mention different examples as motivations for example: multiplying bacteria, radioactive decomposition (exponential growing), the nature of epidemic caused by infectious diseases, the spread of information (logistical growing). What is common in these examples is the following: it is not emphasized enough to get to know the system which is to be modeled measuring has to be done, and these measurements serve as the base of principles with the help of which we can describe the changes. In certain cases it can be reasonable to choose such a phenomenon which can easily be supported by experimental measuring and it is easier to be modeled than the others mentioned above. One of the groups of solid materials is made up by crystalline materials. Beyond the fact mentioned above, the practical importance of this may prove the studying of the process of crystallization.

Figure 1: Crystallization of Sodium Acetate in a test-tube.

2. Description and modeling of phenomenon

Due to certain properties NaAc is especially appropriate to make measuring in connection with the process in order to understand the relation which serves as the base of modelling. We pour the supersaturated solution of sodium acetate ($Na^+CH_3COO^-$ or NaAc) into the test-tube. By adding a piece of crystal we can start the process of crystallization. The speed of change is ideal (not too fast or slow) and the change can be observed well. At the same time the experiment does not require complicated tools and materials. So, this experiment can be carried out even at home. All these facts make it possible to produce measuring of the necessary promptness by using simple tools.

2.1. Mathematical model of one-dimensional case

As we wish to model the process of crystallization it seems to be natural that we consider the amount of substance $X(t)$ (number of moles) as state variable. On the base of experimental measuring we suppose that α quantity of material getting into solid phase during a given time is independent of the quantity of the solid material and the time t:

$$X(t+1) - X(t) = \alpha \quad (t \geqslant 0).$$

By the next point of time the quantity of the solid phase is increased by α. Let h denote the time spent between the two states, so that the problem can be described in a more general way. If we select a longer time interval then more crystals are created and vice versa so it depends on h:

$$X(t+h) - X(t) = \alpha(h) \quad (t \geqslant 0).$$

On the base of experience it is obvious

$$\lim_{h \to 0} \alpha(h) = 0.$$

t_i $[sec]$	0,0	3,0	6,0	9,0	12,0	15,0	18,0	21,0	24,0
$V_i[cm^3]$	0,0	1,3	3,1	4,9	6,5	8,3	10,1	11,9	13,5
$\Delta V_i[cm^3]$	1,3	1,8	1,8	1,6	1,8	1,8	1,8	1,6	-

Table 1: The measured volume of the growing crystal.
$\Delta V_i = V_{i+1} - V_i$ $(i = 0, \ldots, 7)$.

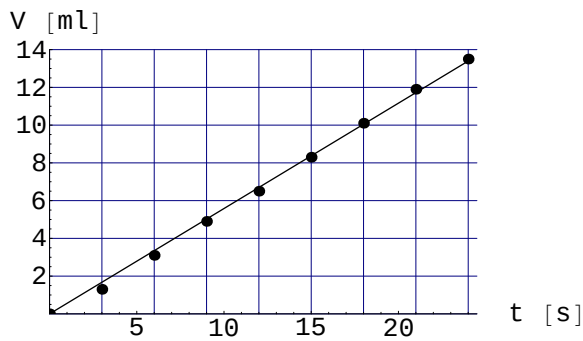

Figure 2: Linear time-dependence of the volume.
(The measured values and the fitted line to them.)

On the base of the Table 1 and the Figure 2 we can assume that there is linear proportionality between h and the increase. During a longer period of time greater quantity of solid material is created. So $\exists \lambda_1 \in \mathbb{R}$ $(\lambda_1 > 0)$, $\alpha(h) = \lambda_1 h$:

$$X(t+h) - X(t) = \alpha(h) = \lambda_1 h,$$

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \lim_{h \to 0} \frac{X(t+h) - X(t)}{h} = \lambda_1,$$

where λ_1 is independent on t and h, only it depends on a characteristic constant of the system. So the phenomenon presented by the experiment can be described by the following differential equation:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \lambda_1 \quad (\lambda_1 > 0).$$

2.2. Exploration of the phenomenon

It is important to note that the speed of the growth of the crystal (the growth of amount of substance of solid phase during a given period of time) depends on the area of the crystal and the concentration of the liquid at a given moment.

During the experiment we produced the supersaturated solution of sodium acetate. Just like other ionic crystals sodium acetate has water molecules bound within its crystal lattice. The quantity of this water characterizes the given ionic crystals. In the case of NaAc 1 mole material has 3 moles water ($NaAc{\cdot}3H_2O \equiv$ sodium acetate trihydrate). The water content of the salt escapes from the lattice during the heating and the material dissolves in this water, that is why the proportion of NaAc and the water is 1 : 3 in the supersaturated solution, too. The process of crystallization is launched by the piece of crystal put into the liquid. During the process the proportion of amounts of substances built into the lattice will be the same, so the concentration of the liquid remains constant. Regardless of the first short period of the process the surface of the increasing crystal which is in contact with the solution, also remains the same. In conclusion, the surface gets forward at an equal speed.

3. Two and three-dimensional extension of model

In the chemical point of view, the same changes can be seen in the case of the well-known hand warmer. (During the heating the supersaturated solution is produced. The launch of the crystallization is caused by the mechanical effect which can be produced by the stainless metal sheet which is in the pad.) In this case if we imagine the pad thin enough, we can idealize the phenomenon that the growth of the crystal is carried out by the following way: starting from a given point of a plain in concentric circles at an even speed. As we know, the speed of the growth depends on the size of the surface of the crystal. In our model it is proportional with circumference

$$K(t) = 2\pi R(t) \tag{3.1}$$

of the circle.

We can interpret the result of experiment that the surface of the crystal moves forward at an even speed where the $R(t)$ root is proportional with the time spent. The move of the surface of the solid material to a given direction of the space is

$$R(t) = \varrho t. \tag{3.2}$$

The quantity of the material built in the solid phase is determined by the radius of the circle during h period of time:

$$\alpha(h) = X(t+h) - X(t) = \beta K(t)h.$$

(In the reality the shape of the crystal can be approached by a cylinder. The surface of its lateral is the product of the height and circumference.) We can take the length into consideration by selecting an appropriate beta constant.

By the relation of (3.1) and (3.2) we can see that $\alpha(h)$ is proportional with the time spent in each moment of time:

$$\alpha(h) = X(t+h) - X(t) = \lambda_2 th \quad (\lambda_2 = 2\beta\pi). \tag{3.3}$$

The expression (3.3) divided by h and h tends to 0:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \frac{X(t+h) - X(t)}{h} = \lambda_2 t.$$

So the two-dimensional growth of NaAc-crystal can be given by the differential equation:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \lambda_2 t \quad (\lambda_2 > 0).$$

If no mechanical obstacle can be experienced in the growth of the crystal it can grow in each direction of the space then we can regard the growing crystal as sphere-symmetric and the speed of the growth is proportional with the surface

$$F(t) = 4\pi R^2(t)$$

of a sphere.

On the base of the facts mentioned above we can get the following differential equation:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \lambda_3 t^2 \quad (\lambda_3 > 0).$$

References

[1] Atkins, P.W., Physical Chemistry I–III., *Oxford Univesity Press*, Oxford (1990).

[2] Borrelli, R.L., Coleman, C.S., Differential Equations: A Modeling Perspective, *2nd Edition, Wiley*, New York, (2004).

[3] Gadomski, G., Siódmiak, J., A Novel Model of Protein Crystal Growth: Kinetic Limits, Length Scales and the Role of the Double Layer, *CROATICA CHEMICA ACTA*, Zagreb (2003).

[4] Geda, G., Vida, J., Observation of mechanical movements through virtual experiments, *6th International Conference on Applied Informatics*, Eger (2004).

[5] Geda, G., Vida, J., Murányi Z., B. Tóth Sz., How to study the phenomena of nature in the future (Multimédia a Kísérleti mérések szolgálatában), *NETWORK SHOP 2005*, Szeged (2005).

[6] Geda, G., Various systems in a single mathematical model, *Teaching Mathematics and Computer Science*, Debrecen (2008).

[7] Hatvani, L., Pintér, L., Differenciálegyenletes modellek a középiskolában, *POLIGON* (1997).

[8] Ponomarjow, K.K., Differenciálegyenletek felállítása és megoldása, *Tankönyvkiadó*, Budapest (1981).

[9] Rontó, M., Raisz, P., Differenciálegyenletek műszakiaknak, *Miskolci Egyetemi Kiadó* (2004).

[10] Tóth, J., Simon, L.P., Differenciálegyenletek, Bevezetés az elméletbe és az alkalmazásokba, *TYPOTEX Kiadó* (2005).

[11] Walter, W., Ordinary Differential Equations, Graduate Texts in Mathematics 182, *Springer, New York* (1998).

Gábor Geda
Department of Computer Science
Eszterházy Károly College
Leányka str. 4.
H-3300 Eger
Hungary

e-mail: `gedag@aries.ektf.hu`

Annales Mathematicae et Informaticae
***35** (2008) pp. 163–172*
http://www.ektf.hu/ami

ICT teaching methods – Programming languages

Zsuzsanna Papp-Varga, Péter Szlávi, László Zsakó

Department of Media and Educational Informatics
Eötvös Loránd University, Budapest, Hungary

Submitted 7 March 2008; Accepted 9 December 2008

Abstract

Today the important ICT topics are taught with the help of various methods. Some of them are unsuitable for successful teaching-learning whereas others may bring about success in certain age groups and class types.

Programming languages were first taught shortly after the appearance of high-level programming languages. First it was done rather as an "art", but later more and more consciously and systematically. However, it should be stated that the methods used in teaching programming languages, as "languages", are far from being near to those of natural languages with respect to their elaborateness, quality and, unfortunately, efficiency.

1. Introduction

The most important teaching methods of the various fields of information and communication technology (ICT) have already been developed [15]. As ICT teaching cannot boast with a long history, in most cases they have not been clearly formulated, and their formation has not been so conscious but rather instinctive, which results in the fact that most teachers do not use one single method but a sort of blend of methods, where one of them is represented dominantly.

This methodological "uncertainty" also ensues that there are teachers who are capable of teaching successfully even when they use a method labelled below as being negative. The negative label can be principally explained by the fact that these methods do not "automatically" ensure good teaching; what is more, it is fairly easy to teach very badly when one relies on them.

Hereinafter the most widespread programming language teaching methods are listed and reviewed:

Statement-oriented (the language is seen as a set of statements, and the individual elements of the set are taught in a certain order).

Using as a tool (when teaching programming and database management, it considers the aspects of database-teaching to be of primary concern, and thus introduces language tools in the necessary extent).

Software technology-oriented (a programming language teaching method adapted to a software developing methodology and technology, where the methodology motivates the choice of a language or even languages).

Task type-oriented (the method is identical with the one discussed at the programming teaching method; it introduces new programming language knowledge in a way that the problems to be solved necessitate).

Language-oriented (the method sees a language as a structural unit, bringing the logic of the language to front, and introducing the concrete elements of a language in the necessary extent and order).

Action-oriented (the statements of a language are taught in a way that it traces them back to an implementation in another language – formerly to assembly statements, now rather to other high-level languages).

Sample task-based (the method presents a language through an analysis of sample tasks).

2. Statement-oriented

The statement-oriented method defines a programming language as a set of statements [1]. It conceives teaching a language as teaching the elements of a set. (And to top it all, in alphabetical order, in the worst case.[1]) The idea of the set also refers to the fact that each element of the set, i.e. each statement of a language must be taught (which leads to a common ICT teaching delusion[2]). Nevertheless, it is easy to foresee the depth of acquiring a language: it is only a mere set of lexical elements lacking any connections among statements or with the modus operandi.

Neither does this notion promote deciding which elements are important and which are not. Moreover, there is no guarantee that one will ever make any use of the elements learned.

If one pictures a language this way, one can claim that a language is an unstructured unity of elements and thus there is no need for any further knowledge to construct programs from the elements: it will develop by "itself".

[1]This idea evolved based on books on programming languages. These books have not yet reached the advanced state of those on natural languages. In most cases the same book is meant to be the manual of a given language (which is practically in alphabetical order), the language coursebook, the dictionary etc.

[2]The notion of popular delusion see in [16].

3. Applying as a tool

There are many programming teaching methods where program writing is more or less an automated activity, and can be done with the help of coding rules and coding conventions. In this case the programming language appears as a result of the coding process. One always needs only that amount that one needs for coding one's algorithms [17, 13].

Let us see some examples for the above (Pascal encoding rules):

Algorithmic statement (with Hungarian keywords)	**Pascal code equivalents**

Be: variables [conditions]	**Repeat**
	  write('question?');
	  readln(variables)
	until conditions;

Ha conditions **akkor**
  statements
 	
különben
  statements
Elágazás vége
	If condition **then begin**
	  statements
	end
	else begin
	  statements
	end;

Ciklus amíg condition
  statement
Ciklus vége
	While condition **do begin**
	  statements
	end;

When using this method, it is guaranteed that the acquired language elements will later be used again. Since the structures, algorithmic elements and data types recur regularly in programming craft, one can also state that the acquired elements have to be often used.

4. Software technology-oriented

Relying on the above principle, Tibor Temesvári has constructed object-oriented programming (OOP) and its implementation in the Pascal and C++ programming languages. First, he discusses object-oriented programming in general (1. Characteristics of OOP, 1.1. Classes and Objects, 1.2. Encapsulation, 1.3. Inheritance, 1.4. Implementation of Inheritance, 1.5. Using Inheritance, 1.6. Multiple Inheritance, 1.7. Type Compatibility, 1.8. Polimorphism, 1.9. Dynamic Binding, 1.10. Virtual Method, 1.11. Execution of Methods 1.12. Object-oriented Programming Languages), in which he does not touch upon concrete programming language knowledge, but only deals with the object-oriented technology.

In Chapter 2 the above are followed by teaching the implementation possibilities i.e. programming language skills (2. OOP in Pascal, 2.1. Planning, 2.2. Defining a Class 2.3. Interface Part, 2.4. Implementation Part, 2.5. Self, 2.6. The Declaration of Objects, 2.7. Using Objects, 2.8. Inheritance, 2.9. Procedure Calls Defined

in the Ancestor, 2.10. Redefined Methods, 2.11. Virtual Method Table – VMT, 2.12. Constructors (procedures), 2.13. Dynamic Methods, 2.14. Dynamic Method Table (DMT), 2.15. Type Compatibility, 2.16. Dynamic Objects, 2.17. Cleaning up Dynamic Objects, Destructors). [18]

Similarly, some software technology (the OOP, database management, COM- and web-programming) denotes the guideline of Delphi language processing in a book by Marco Cantù: [8]. There is also a good example for this in the topic of web design in a book by Kris Jamsa et al. [4]. New paradigms including aspect-oriented and generic programming may also affect teaching programming languages. [10]

5. Task type-oriented

In this case the elements of a programming language are introduced because they are needed in the process of problem solving. The various elements do not turn up because some educational objective requires them, but because the next task cannot be solved without them. [12, 19]

The task below comes from a class introducing PROLOG that we developed (relying on Turbo PROLOG system):

Step 1: facts

```
one'sfather(father,child).
one'smother(mother,child).
```

Step 2: clauses

```
one'sparent(X,Y) if one'smother(X,Y).
one'sparent(X,Y) if one'sfather(X,Y).
```

Step 3: **or** operation in clauses

```
one'sparent(X,Y) if one'smother(X,Y) or one'sfather(X,Y).
```

Step 4: **and** operation in clauses

```
one'sgrandparent(X,Y) if one'sparent(X,Z) and one'sparent(Z,Y).
```

Step 5: recursion in clauses

```
one'sancestor(X,Y) if one'sparent(X,Y)
                   or one'sparent(X,Z) and one'sancestor(Z,Y).
```

Step 6: "`any`" value in the place of parameters

```
parent(X) if one'sparent(X,_).
```

Step 7: "`not`" operation i.e. negation in clauses

```
notparent(X) if one'sparent(_,X) and not (one'sparent(X,_)).
```

Step 8: cut operation in clauses

```
oneparent(X) if parent(X) and !.
```

Step 9: display, and equally false formula in clauses

```
allparent if parent(X) and write(X) or fail or nl.
```

Step 10: equivalency check in clauses

```
twochilded(X) if one'sparent(X,Y) and one'sparent(X,Z) and
                 not (Y=Z).
```

Step 11: new programming skill without new language element

```
onechilded(X) if one'sparent(X,_) and not (twochilded(X)).
```

Similar examples can be found in the syllabus on teaching Logo programming language developed at Eötvös Loránd University. Its subjects and the new language elements to be learned in brackets are as follows:

- Drawing elementary shapes (forward, back, right, left, repeat)
- Constructing from shapes (learn, penup, pendown)
- Principles of making complex figures
- Circles, arcs (setpencolor, setpenwidth!)
- Recursion, trees (if)
- Line patterns, shape patterns (fill, setfillcolor, setfillpattern)
- Logo and the frame of reference (setx, sety, setheading)
- Fractals

An important development in teaching programming languages is that the statement-oriented method is often merged with this one [20], since the abstract, "crystal clear" know-how of the previous notion, which is free from programming problems, are completed with the real-life experience of statements. It is the way that makes the level of language teaching rise significantly!

6. Language-oriented

The language-oriented variant regards the language as a *structured unit*. It examines the calculation model belonging to the language [2] (in primary and secondary education only von Neumann-principled, automaton-principled, functional, and logical languages can be present). Then it reviews the main framework of the build-up of programs e.g. in the Pascal language:

```
Program name;
  declarations
begin
  statements
end.
Declarations: label definitions
              constant definitions
              type definitions
              variable declarations
              procedure and function definitions
```

Becoming familiar with the basic concepts used in a programming language and their possible implementations in that language (e.g. compilation unit, program unit, block structure, memory management, declaration evaluation, concepts regarding variables, concepts regarding type, parameter pass etc.) also tightly belong to the build-up of a program [18].

The next step might be that certain elements of the language are examined and it is given how the programming structures are implemented in that given language. For example as for Pascal, one might claim the following about conditional loops (before setting the concrete syntactic rules):

The Pascal language can have pre-test or post-test conditional loops. For pre-test loops the condition is first evaluated – if the condition is true, the code within the block is then executed. This repeats until the condition becomes false. On the other hand, for post-test loops the exit condition must be set. The core of a pre-test loop can be one single statement. If more statements are necessary, they must be surrounded by statement brackets. Contrarily, the core of a post-test loop can contain any number of statements.

Finally, only after the above can one give the syntax and semantics of the statements. As opposed to higher education, in primary and secondary education it is usually not a formal method that is used but a demonstration via examples. To define syntax, only the format of the statement is given (pl. **while** `condition` **do** `statement`). For semantics, however, smaller programs are used, through which the operation of the given statement can be understood (with the help of the method described in the next chapter).

It should be noted that in higher education this method is becoming more and more widespread in demonstrating the possible elements of a programming language, bringing examples parallelly from several languages [9, 18]. For instance, the course *Functional languages*, taught by Zoltán Horváth at bachelor's degree courses for programmers at the Faculty of Informatics, Eötvös Loránd University, follows the same structure. Of course, both the objective and the presupposed basics are different from those in primary and secondary education.

7. Action-oriented

Here the primary criterion is to understand how the statements operate i.e. to make students able to visualize what happens when the statements are being executed. In the simplest case one can give the statements of the language in another known language, perhaps in assembly language.

The example explaining `DO` statement below comes from a classical FORTRAN coursebook [6]:

```
     K=1
17   T=0.0
     J=1
18   T=T+A(I,J)*B(J,K)
     J=J+1
     IF(J-N)18,18,20
20   C(I,K)=T
     K=K+1
     IF(K-N)17,17,21
21
```

```
     DO 20 K=1,N
     T=0.0
     J=1
18   T=T+A(I,J)*B(J,K)
     J=J+1
     IF(J-N)18,18,20
20   C(I,K)=T
21
```

In a book on C# one can read the following explanation about the ++ operator [3]:

```
a = ++b;   // ⇔ b = b+1; a = b;
```

```
a = b++;   // ⇔ a = b; b = b+1;
```

The above examples show that one should not necessarily go back to another language to describe the operation of a statement, but one can define it with the help of other elements of the same language.

In a sense, a possible solution belonging here is when the semantics of the elements of a new programming language is defined with the elements of a "well-known" algorithmic language. This results in an extra educational profit: while the elements of a language are demonstrated, the students can practise programming in an algorithmic language, as well.

8. Sample task-based

According to this notion, if students are shown quite a lot of examples, they will be able to acquire a programming language well [7]. Here is a quotation from a book by Zsuzsanna Márkusz called "It is Easy to Write PROLOG Programs":

"*My PROLOG teaching experience has convinced me that the easiest way to learn programming is through sample programs. Therefore, instead of any scientific introduction (notations, definitions, theorems) the book foreshows twelve sample programs, which are explained in great detail.*"

Although it shows some resemblance to the task type-oriented method, their basic principles are different. There the root of the matter is that the set of tasks makes it necessary to introduce new language elements. As for this method, it is just the opposite: the language elements are given in the tasks and their build-up follows

the language elements. That is why it is not certain that the acquired language elements will need to be used in the future and they might be forgotten if not practised.

8.1. A short evaluation of the above methods

We think that the *statement-oriented* method is unsuitable for teaching programming languages because a programming language is not equal to a set of statements. Behind a programming language there is always an idea, and in order to apply a programming language properly, it is inevitable to learn it[3]. Programming languages use basic language concepts like type, block structure, parameter pass etc., which might be different in various language types, or even in languages and their knowledge is connected primarily not to statements, but to languages. In each programming language a program has some structure, some constructing rule.

The *"Applying as a tool" approach* is the one that is needed in algorithm- and data-oriented teaching of programming, and thus this method can be used parallelly with the above programming teaching methods, that is with teenage students considering ICT as a carrier.

The *software technology-oriented* method is, actually, an improvement of the previous one (applying as a tool) for higher education, ICT specialists' education; so it can be a very powerful method there. On the other hand, in primary and secondary education it could have a role maximum in ICT vocational training.

The *task type-oriented* method is the only one that can be used in each level of primary and secondary education, where the main objective is the implementation and try-out of algorithms, and not a thorough knowledge of a programming language (quotation from the justification in the Hungarian National Curriculum: *It is enough to teach a programming language to that depth that is necessary for implementing and trying out algorithms. The language itself is not a crucial part of the ICT curriculum.* [21]).

The *language-oriented* concept may be excellent to summarize the elements of a language as completing language learning. For those considering information science as a carrier, it is also possible to introduce a new language that is fairly similar to the ones learnt before (e.g. after Pascal Delphi, C++ can be taught this way; or after C++, C#, ...), as in this case the students' previous language knowledge can be used effectively.

The *action-oriented* idea greatly resembles to the statement-oriented one, since it teaches statements, as well. On the other hand, here the definitions are given on the level of the "operation" of statement (i.e. how they work) instead of their "specification" level (i.e. what they should do). If used exclusively for beginners, it will not bring any success.

Teaching with the help of *sample tasks* is a "medieval" concept. This way one can train "artists" of programming and not its conscious doers.

[3]It is the "cost" of how to discover this world of ideas that qualifies the language itself. That is why it has such special importance in education. [14]

Note. The presentation of language teaching methods is somehow dangerous when relying on books on programming languages. The reason for this is that today the methodological background of books on programming languages is much weaker than that of those on natural languages. That is why the same book discusses a language in several ways from several aspects.[4]

References

[1] Alcock, D., Illustrating BASIC! *Cambridge University Press*, 1977. (Hungarian translation: Ismerd meg a BASIC nyelvet! *Műszaki Könyvkiadó*, 1984.)

[2] Horowitz, E., Fundamentals of programming languages, *Springer Verlag*, 1983. (Hungarian translation: Magasszintű programnyelvek, *Műszaki Könyvkiadó*, 1987.)

[3] Illés Z., Programozás C# nyelven, *Jedlik Oktatási Stúdió*, 2004.

[4] Jamsa, K., Lalani, S., Weakley, S., Web-programming, *Jamsa Press*, 1996. (Hungarian translation: A Web programozása, *Kossuth Kiadó*, 1997.)

[5] Lischner, R., Delphi in Nutshell, *O'Reilly & Associates*, Inc., 2000.

[6] Lőcs Gy., Vigassy J., A FORTRAN programozási nyelv, *Műszaki Könyvkiadó*, 1973.

[7] Márkusz Zs., PROLOG-ban programozni könnyű, *Novotrade*, 1988.

[8] Cantù, M., Mastering Delphi 5, *Sybex*, 1999.

[9] Nyékiné Gaizler J. (szerk.), Programozási nyelvek, *Kiskapu Kft.*, 2003.

[10] Porkoláb Z., Kozsik T., Zsók V., Új szoftverparadigmák nyelvi támogatása: a jelen oktatása – a holnap technológiája, *Informatika a felsőoktatásban'05. Debrecen*, aug. 24–26, 2005. `http://aszt.inf.elte.hu/~fun_ver/2005/papers/if05_paper_zsv.pdf`

[11] Stroustrup, What is "Object-Oriented Programming"? (1991 revised version), *Proc. 1st European Software Festival*, February, 1991. `http://www.research.att.com/~bs/whatis.pdf`

[12] Szentpéteriné Király T., Comenius Logo teknőcgrafika, *Kossuth Kiadó*, 2000.

[13] Szlávi P., A programkészítés didaktikai kérdései, *PhD thesis*, 2004.

[14] Szlávi P., Programozási nyelvek értékelése, *electronic script*, 1999. `http://digo.inf.elte.hu/~szlavi/InfoOkt/SzoftErt/Szoftverek%20értékelése.pdf`

[15] Szlávi P., Zsakó L., Methods of teaching programming, *Teaching Mathematics and Computer Science 1*, No. 2 (2003) 247–258.

[16] Szlávi P., Zsakó L., Delusions in informatics education, *Teaching Mathematics and Computer Science 2*, No. 1 (2004) 151–152.

[17] Szlávi P., Zsakó L., Temesvári T., Módszeres programozás: A programkészítés technológiája, *ELTE IK*, 2007.

[4] In most cases the same book can be used as a manual and a coursebook of a programming language. Moreover, it often contains programming know-how. Now let us compare them with books on natural languages, where one can find monolingual dictionaries, bilingual dictionaries, coursebooks, workbooks, books to develop speaking skills etc.

[18] Szlávi P., Zsakó L., Temesvári T., Programozási nyelvi alapfogalmak, *ELTE IK*, 2005.

[19] Turcsányiné Szabó M., Zsakó L., Comenius Logo gyakorlatok, *Kossuth Kiadó*, 1997.

[20] Végh Cs., Juhász I., Java – start! *Logos* 1999, 2000.

[21] Zsakó L., Az informatika ismeretkörei, *ELTE IK*, 2005.

Zsuzsanna Papp-Varga
Péter Szlávi
László Zsakó
Department of Media and Educational Informatics
Eötvös Loránd University
Budapest
Hungary

e-mail:
vzsuzsa@elte.hu
szlavi@ludens.elte.hu
zsako@ludens.elte.hu

Annales Mathematicae et Informaticae
***35** (2008) pp. 173–188*
http://www.ektf.hu/ami

The evolvement of geometrical concepts in lower primary mathematics

(Parallel and Perpendicular)

Ibolya Szilágyiné Szinger

Eötvös József College, Baja, Hungary

Submitted 17 March 2008; Accepted 10 July 2008

Abstract

The evolvement of some geometrical concepts, such as parallel and perpendicular has been studied in a developmental teaching experiment in class four whose aim was to put the model of geometry teaching according to van Hiele into practice.

Our research question is how lower primary geometry teaching in Hungary, particularly teaching the concept of parallel and perpendicular is related to the levels formulated by van Hiele. Moreover to what extent are the concrete activities carried out at these levels effective in evolving the concepts of parallel and perpendicular.

Our hypothesis is that in lower primary geometry teaching (classes 1–4) only the first two stages of the van Hiele levels can be put into practice. By the completion of the lower primary classes transition to level 3 is not feasible. Although the set of concepts are evolved but there is not particular relationship between them. The logical relationships between the characteristics of a shape are not really recognized by children. They are not able to infer from one characteristic of a shape to another.

Keywords: mathematics teaching, parallel, perpendicular

MSC: 00A35, 97D70, 51F20

1. Introduction

Teaching geometry in Hungary in the first four grades of primary school aims at laying the ground to establish the skills through which learners can prepare for gaining knowledge on their own.

The basis of learning geometry is inductive cognition based on gaining knowledge. Starting out from the concrete and gathering experience from various activities will finally lead to the formulation of general relationships. The third educational principle laid down by Farkas Bolyai also emphasizes the importance of starting with the concrete: "(The teacher) ... should always start with what learners can see and touch, and not with general definitions (it is not grammar that the first utterance is based on) and he should not torture prematurely with longwinded reasoning... We should start with geometric shapes and reading... and we also should get out of the sheet..." (Dávid, 1979)

In the framework curriculum what is particularly highlighted is the development of orientation in plane and shape, the formation of geometrical knowledge through recognizing geometrical shapes and the characteristics of forms and quantity as well as simple transformations.

In lower primary the basis of mathematical concepts is laid down. In this paper the development of the concepts of parallel and perpendicular is examined. In the framework curriculum the requirements of the teaching material related to parallel and perpendicular are as follows:

Grade 3: Measuring angles with non-standard units.

Grade 4: Producing plane figures by means of pairs of parallel and perpendicular straight lines. Measuring angles with right angle, its half and its quarter.

In some course books the concepts of parallel and perpendicular are introduced in various grades and in a different way. In a course book for grade 4. by C. Neményi it is during the presentation of the opposite and neighbouring faces of the rectangular solid and the cube that the concepts of parallel and rectangular are introduced and then they move on to the plane. In the course books by Hajdu, Török and Rakos these concepts are introduced in grade 3 when the position of pairs of straight lines in plane are studied. In the book entitled "Colourful mathematics" it happens in the same way, but only in grade 4.

The evolvement of some geometrical concepts such as parallel and perpendicular has been examined in an educational development experiment conducted with grade 4 pupils. The results and the lessons of the experiment are described below.

1.1. Linguistic problems

"One of the most important objectives of school is to provide children with a means i.e. language through which they will be able to learn, to think and speak about the world in which they actually live. Or to be more precise we intend to assist them to make this means they already possess more sophisticated." (Holt, 1991)

Several children, like parrots, are able to repeat sentences containing technical terms, but they have not the slightest idea about their meaning. As a matter of fact this is what we would like to avoid. Some children on the other hand clearly understand the mathematical concepts and problems but they are at a loss when they should express their knowledge and thought in proper terms. Every one needs some time to incorporate words describing concepts into their active vocabulary and

they will be able to express their thoughts by means of proper terms. "Teachers can assist the process of incorporating words into learners' active vocabulary in a way that they will not correct the improper language use all the time, moreover they try to understand children's any, even inaccurate utterances in order that children could be able to become aware of their observations. But when the teachers want children to use a 'technical term' they have already understood, it is practical if they themselves repeat the utterance using the proper words." (Szendrei, 2005)

According to Andre Revuz in every field of science, including mathematics, the most fearful obstacle to understanding is the language barrier.

What kind of context is accessible to fourth grade learners? Is the language of mathematics course books for lower primary learners suitable for the teaching material to be acquired?

Here is a selection of sentences related to parallel and perpendicular from mathematics course books for grade three and four:

"Parallel straight lines do not share any point, their position and distance is the same everywhere." (Török, 2002)

"The length of perpendicular sections drawn between parallel straight lines is the distance between parallel straight lines." (Hajdu, 2005)

"The pairs of straight lines in plane which do not have a point in common, no matter how much we make them longer , are called parallel straight lines."

"Four angles made by two perpendicular straight lines are equal." (Árvainé, 2005)

"The distance between parallel straight lines is everywhere the same."

"Draw straight lines in a way that draw the lines along the two edges of the square ruler that make a right angle." (Rakos, 2002)

In the framework curriculum the proper and exact level of mother tongue and technical terms suitable for the age group is emphasized. We wonder whether the above sentences are suitable for the language use of learners.

2. Theoretical background

2.1. The levels of geometrical thinking according to van Hiele

Young children start gaining knowledge in geometry already in kindergarten where the concept of geometrical objects (geometrical solids, plane figures) is being established by examining the shapes of the objects in the environment. Establishing the characteristics for the set of these objects implies a higher degree of gaining knowledge. A large amount of references can be found on gaining geometrical knowledge, but in this particular case we rely on van Hiele.

According to P-H van Hiele the process of gaining geometrical knowledge can be divided into five levels.

At the level of global cognition of shapes (level 1.) children perceive geometrical shapes as a whole. They easily recognize various shapes according to their forms, they remember the names of the shapes however they do not understand the

relationship between the shape and their components. They do not recognize the rectangular prism in the cube, rectangle in the square, because these are totally different things for them.

At the level of analysis of shapes (level two) children break shapes down into components and then put them together. They also recognize the faces, edges and vertices of geometrical solids as well as the plane figures of geometric solids which are delineated by curves, sections and dots. At this level particular importance is attached to observation, measurement, folding, sticking, drawing, modelling, parquetry, and using mirrors. By means of these concrete activities children can establish and enlist the characteristics of shapes such as the parallelism, perpendicular of faces and sides, characteristics of symmetry, the presence of right angle etc, but they are not able to define and to recognize the logical relationships between the characteristics. At this level children do not perceive the relationships between shapes.

At the level of local logical arrangement (level 3) learners are able to find relationships between the characteristics of a particular shape or between various shapes. They can also make conclusions from one characteristic of shapes to the other. They understand the importance of determination, definition. However the course of logical conclusions is set by the course book or the teacher. The need to prove things is started, but it applies only to shapes.

Level four (making efforts to reach complete logical set-up) and level five (axiomatic set-up) belong to the requirements of secondary and tertiary education.

In the van Hiele model each learning stage is constructed and enlarged by the thinking established by the previous stages. Transition from one level to the other happens continually and gradually, while children are acquiring the mathematical terms according to the particular levels. This process is particularly influenced by teaching, especially its content and method. For the suitable geometrical thinking none of the levels can be omitted. Every level has its own language, system of notation and logical set-up. From educational point of view it is highly relevant in the theory of van Hiele that we cannot expect from learners at a lower stage to be able to understand the instructions formulated in terms of a higher level. According to van Hiele this is probably the most frequent reasons for failures in mathematics teaching.

2.2. Concept formation

During the formation of a mathematical concept, the concept has to be fitted into the system of concepts established before (assimilation) but it can happen that the modification of the existing system or pattern is necessary for the fitting of the new concept. The balance of assimilation and accommodation is absolutely indispensable for the proper formation of concept. If this balance is upset i.e. assimilation is not followed by accommodation then the learners' own interpretations find their way into their mathematical knowledge, which later on may lead to misconceptions. Then the concepts formed in this way can be vague and inaccurate.

Teaching geometric concepts is as a matter of fact a long process. The principle

of progressiveness should be observed, and accurate definitions should be established but not by all means. Sometimes even at lower primary classes definitions are provided in course books, although learners lack the required experience and abstraction level. In this respect, what R. Skemp the mathematician and psychologist said is:

"By means of definitions it is impossible to transmit concepts to anyone which are at a higher level than his knowledge, only by providing plenty of proper examples. Since in mathematics these examples mentioned above are almost all of them various concepts, therefore we have to make sure that the learners have already acquired there concepts. Selecting the proper examples is a lot more difficult than we suppose. The example should possess those common characteristics which make up the concepts, but they should not have any other common characteristics." (Skemp,1975)

The evolvement of scientific concepts, such as parallel and perpendicular is based on education. According to his observations Vigotsky came to the conclusion that "in as much as the progress of teaching contains the proper elements of the curriculum, the development of scientific concepts will proceed the development of spontaneous concepts." In the progress of teaching the special co-operation of children and adults and the transmission of the teaching material in an order can give an account for the premature achievement of concepts. According to teaching experience it can be understood that the direct teaching of concepts is not really possible. The mere acquisition of a new word verbally covers only emptiness. In this case children acquire only words and not concepts. When children first recognize the meaning of a new word then the process of evolvement of a concept is being started. Scientific concepts are not acquired and learned "ready-made" by children but these concepts are evolved and established through the active thinking of children. The evolvement of spontaneous and scientific concepts is closely related to each other. A basic requirement for the evolvement and acquisition of scientific concepts is the proper level of spontaneous concepts. However the evolvement of scientific concepts can also have an influence on the development of spontaneous concepts.

Bruner's representation theory is also based on activities: According to this theory in order that learners could understand the teaching material, they should "process" it intuitively before. According to Bruner every process of thinking can happen on three levels:

a) enactive level: gaining knowledge through concrete practical activities and manipulations.

b)iconic level: gaining knowledge through graphic images, imaginary situations.

c) symbolic level: gaining knowledge through mathematical symbols and language.

Although in the lower primary grades the enactive and iconic levels are in the foreground, but language (speech), which is the symbolic level, is also very important. If we get round the first two levels there is a risk that learners due to the lack of proper system of images will not be able to solve mathematical problems and to

understand concepts at symbolic level, because they have nothing to rely on. If the first level (the concrete, practical activities) is omitted, then the proper system of images will not be established.

"The concept image is the total cognitive structure associated with the concept name, which includes all the mental pictures and associated properties and processes, pictures, graphs experiences". (Tall, 2004)

In our teaching experiment learners have gained a wide range of experience of the concepts of parallel and perpendicular through various concrete activities such as modelling, folding, clipping and drawing. Thus, their concept image will be versatile.

3. A developmental teaching experiment

3.1. Research question

The research question raised is what the relationship is between lower primary geometry teaching including the teaching the concept of parallel and perpendicular and the geometric levels according to van Hiele. Furthermore how efficiently the concrete activities at these levels contribute to the establishment of the concept of parallel and perpendicular.

3.2. Hypotheses

In lower primary (grades 1–4) geometry teaching can reach the first two stages of geometric thinking according to the van Hiele levels. It is not feasible to reach level 3 by the completion of lower primary. Although sets of concepts are established, but there is no relationship whatsoever between them. Actually children do not recognize the logical relationships between the characteristics of a shape and they are not able to draw conclusions from one characteristics of a shape to another.

3.3. Research methodology

It was in May–June 2006 that the teaching experiment was carried out whose content and method was devised by the author, who also was involved in the lessons. The teacher was a mathematics teacher and supervisor in class 4.c of the Practice School of József Eötvös College in Baja, whose job was assisted by the author in presenting, modeling and eventually raising supplementary questions or alternative explanations. Both of them helped the pupils in carrying out and solving the tasks in individual or pair work. Both checking and evaluation were done in cooperation. During the developmental teaching the evolvement of several geometrical concepts such as rectangles, squares, parallel, perpendicular and symmetry was examined but here we are going to present the formation of the concept of parallel and perpendicular.

The developmental teaching experiment included 16 lessons and the aim was to put the van Hiele model of geometry into practice. In the first lesson a pre-test was done by 26 pupils of class 4 so that we could see that the transition from level 1 (the global recognition of shapes) to level 2 (the analysis of shapes) and the further development of geometric thinking is feasible. When compiling the pre-test, the syllabus of class 3 and the comments of the mathematics teacher were taken into consideration. The first lesson of the development teaching experiment was also the first lesson of the geometry topic as well.

3.4. Pre-test

The task of pre-test 3.was to reveal the conceptual level of the right angle and the angle smaller and bigger than right angle. The task for the children was to decide about angles of seven plane figures as to which of them are right angles, smaller or bigger than right angle and they coloured the angles red, blue and green respectively.

The angles of the plane figures below were examined by the children:

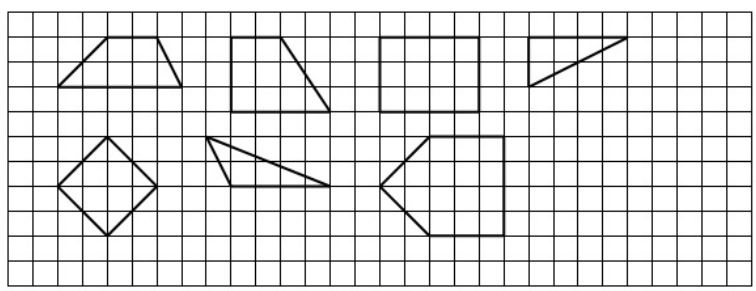

Figure 1: Pre-test

The results are shown in the following table:

The size of every angle is correct.	30.8%
Every right angle was marked properly.	46.2%
One mistake in finding right angles.	15.4%
Right angles were marked properly only in squares end rectangles.	26.9%
Every acute angle and obtuse angle was properly coloured.	30.8%
Two or more mistakes were made in finding acute angles and obtuse angles.	69.2%

As it is revealed by the data the concept of angle needs further development. Almost 70% of the learners made mistake in establishing the size of angles. They proved to be most successful in designating right angles, almost half the group came up with the right solutions. However we still cannot be satisfied with this result.

Tasks related to parallel and perpendicular were not given since children were not familiar with these concepts.

3.5. A developmental teaching experiment

When compiling the teaching material the principle of gradualness was observed and the problems were made more and more difficult. During the first lessons we focused on the characteristics of rectangular solids and cubes. When examining the position of opposite and neighbouring faces the new concept of parallel and perpendicular were introduced. In case of various solids the position of the opposite and neighbouring faces were examined then after spreading the solids we moved on to the plane. In the plane first learners came across with parallel and perpendicular when studying the opposite and neighbouring sides of rectangular and squares.

The detailed description of the lessons can be found in the supplement.

When designing the lessons what we considered of utmost importance was that children could discover geometrical concepts first through concrete experience in real games and activities, later at visual level (drawing) then at an abstract level.

3.5.1. Concrete, manual activities

a) Showing parallel/non-parallel using both hands in various positions.

b) Showing perpendicular/non-perpendicular using both hands in various positions.

c) Producing perpendicular/non-perpendicular position with the leaves of the course book.

d) Finding the opposite and neighbouring faces of the regular pentagonal prisms and square based pyramids, studying their position from the point of view of parallel and perpendicular.

e) Producing plane figures from two coincident right angled triangles. Studying the parallel and perpendicular opposite and neighbouring sides of the quadrangles produced in this way.

f) Demonstrating parallel and perpendicular pairs of straight lines in plane by means of two skewers, also demonstrating non-parallel and non-perpendicular straight lines.

Etc.

Minutes were taken of every lesson, in which the children's responses were also recorded. When examining the position of the faces of the regular pentagonal prism the following conversation took place:

-Show me please which face is opposite this lateral face? What do you think, Petra?

-This one and that one. (She pointed at the two non-neighbouring faces.)

-Are the opposite faces parallel or not?

-No, they aren't.

-Why not?

-Because they are slanting...

-Are these two neighbouring lateral faces parallel or not? (The teacher pointed at them.)

-They are not parallel because they meet.
-Are they perpendicular to each other?
-No, they aren't.
-Why not?
-Because they do not make a right angle, I have checked it with a folded right angle.

In task f) when demonstrating the parallel position, after considering several solutions, Szabolcs came up with the following statement: "It did not matter either how far the skewers were from each other."

Beside the examples demonstrating the concept, examining counter examples is also essential in order to establish a clear-cut concept. After analyzing several examples and counter examples children will reach the level, where they will be able to recognize the essential characteristics of a concept and they will able to differentiate between the essential and the irrelevant characteristics.

3.5.2. Visual tasks

a) Drawing parallel and perpendicular pairs of straight lines on grid.
b) Drawing various triangles on grid and colouring the sides perpendicular to each other.
c) Colouring the parallel lateral pairs of various plane figures, designating right angles.
d) Drawing quadrangles of given characteristics.
e) Sorting out plane figures according to given characteristics.
Etc.

In task c) when colouring the parallel sides of a general trapezoid, the following conversation took place between the author and the child called David:
"You haven't coloured anything in this quadrangle. Haven't you found parallel straight lines?"
"No, I haven't."
At this point the teacher in order to help the child placed the skewers on the two bases, thus David could see that they do not meet. David made the following remark:
"It is true that the skewers do not meet, but the sides are not of equal length, thus they cannot be parallel."
"But this was not a condition for parallel."
Then David said:
"Well, in this case, they are parallel."
Then he corrected the mistakes, which may have come from the fact that most of the time was devoted to the characteristics of square and rectangle. In these quadrangles beside the fact the opposite sides were parallel, they were also of equal length. David probably connected these two characteristics.

3.5.3. Abstract level

After gaining experience at the previous two levels the characteristics of various geometrical shapes were summarized at an abstract level:

-in case of solids, especially cubes and rectangle prisms counting the number of faces, edges and vertices repeatedly, determining the length of edges, the parallel and perpendicularity of faces and edges, and the number of symmetry planes;

-in case of polygons, especially squares and rectangles counting the sides and vertices repeatedly, examining the length and parallel and perpendicularity, determining the number of symmetry axes, and the size of angles produced by the neighbouring sides. Obviously the geometric characteristics were studied through models or the visual representation of the given shape.

Twenty questions is one of the favourite games among children, which is also suitable for practising the characteristics of solids and plane figures. During a game what the children had to guess was the rectangle. These are the questions of a game:

-Does it have five vertices?
-Is it a quadrangle?
-Does it have a right angle?
-Are the opposite sides parallel?
-Does it have a symmetry axis?
-Does it have two symmetry axes?
-Are the sides of the same length?
-Does it have perpendicular sides?

At this point the teacher said they could have guessed it from an earlier question.

-Does it have several right angles?
-Does it have four right angles?

Finally they found out what it was.

During the game of twenty questions we wanted the children to realize that instead of just guessing it is a good strategy to limit the options. We had to convince them that they should not be afraid of asking questions and they should also see that some questions are more purposeful than others and "no" to a good question is just as good as a "yes". Moreover it is no use asking a question when they already know the answer.

3.6. Post-test

The developmental teaching experience was completed by an evaluation worksheet, which was filled in by 25 learners in class 4.c, 23 learners in class 4.a and 24 learners in class 4.b respectively. In these latter two classes the mathematics teacher - supervisor was another teacher.

In the worksheet it was only the questions related to the establishment of the concepts of parallel and perpendicular that were evaluated.

In the task of recognizing the parallel and perpendicular lateral pairs of faces of polygons, children were asked to examine nine shapes. They also did the following

tasks:

a) colouring the parallel sides using the same colour;

b) colouring the right angles red;

c) enlisting the letters designating the plane figures which have parallel sides;

d) enlisting the letters designating the plane figures which have perpendicular sides.

The following polygons have been examined:

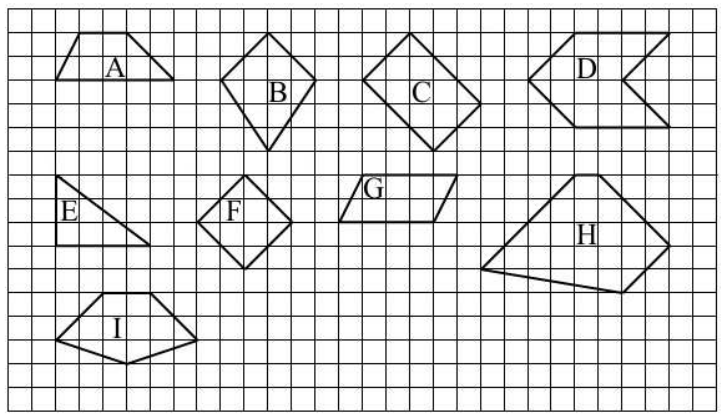

Figure 2: Post-test

The results of the tasks are shown in the chart below:

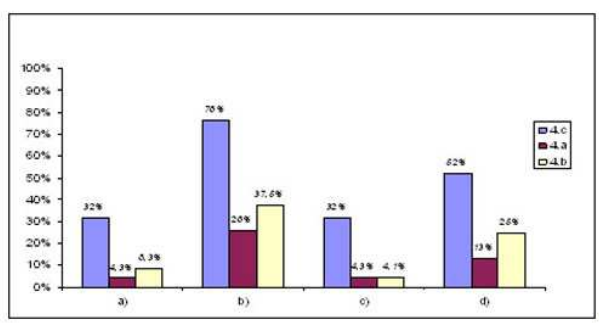

Figure 3: Correct solutions

In the experimental group children's best results were gained in colouring right angles. It was in grade 3 that they came across this concept and it was further developed in grade 4. The outcome of the experiment is shown by the fact that the number of learners who correctly marked the right angle increased from 46% in the pre-test to 76% in the post-test. The results in the control group were significantly

worse, 26% and 37% respectively. The difference between experimental and the control group was also considerable in the recognition of parallel and perpendicular lateral pairs.

In the task of the worksheet related to the characteristics of squares and rectangle children were asked to underline the statements which were true for

a) squares:

Its opposite sides are parallel.
Its opposite sides are perpendicular.
The neighboring sides are parallel.
The neighboring sides are perpendicular.
Every angle is right angle.
Not every angle is right angles.
It has exactly two symmetry axes.
It has four symmetry axes.
It has 8 symmetry axes.
Its every side has same length.
Its opposite sides have same length.

b) rectangles:

Its opposite sides are parallel.
Its opposite sides are perpendicular.
Its neighboring sides are parallel.
Its neighboring sides are perpendicular.
Its angles are all right angles.
Its angles are not all right angles.
It has four symmetry axes.
It has four symmetry axes.
The diagonals are symmetry axes.
Every side has same length.
The opposite sides have same length.

In the evaluation of the tasks we have focused on only the results without any mistake. In the experimental group all the true statements related to the properties of the square were chosen by 52 of the pupils, whereas in the control group the results were 35% and 42%. In case of rectangle the results were as follows: In the experimental group it was 64% in the control groups 39% and 70%. The mistakes can be due to the misunderstanding of the terms 'opposite' and 'neighboring' on the one hand and in the fact the concepts of parallel and perpendicular were not firmly established.

During the developmental experiment only the initial steps were taken to establish the concepts of parallel and perpendicular.

4. Conclusion

The developing teaching experiment guided by the author efficiently contributed to the establishment of the concept of parallel and perpendicular, and the comparison of the results of the pre-test and the post-test also supported the above finding. In the experimental group the results were significantly better than in the control groups. Our findings are related to only the samples examined, which are not representative, and therefore no statistical trials have been carried out. The data measured, the interviews and the games support the hypothesis that it is not possible to reach level 3 of geometrical thinking according to van Hiele by the completion of lower primary (the first four classes of primary education), only reaching the first two levels is feasible. Children are not really able to make conclusions from one characteristic of the figures to the others. They cannot find the relationships between the characteristics of a given figure.

The cognition of children of 6–10 years olds is highly attached to real life, which is why during the formation of concepts only starting out from concrete, manual activities and examples taken from their immediate experience is it possible to reach the level of abstraction. A large number of examples and counter example and making the concept concrete several times and modelling are the preconditions that make it possible for children to recognize the essential characteristics of a concept and they could reach the level of abstraction.

As György Pólya said: "We should not pass up anything that could bring mathematics closer to students. Mathematics is a very abstract science and this is why it has to be presented in a very concrete way." (Pólya, 2000)

5. Supplement

Lesson 1: Pre-test.

Lesson 2: Naming and describing rectangular objects, such as matchboxes, cupboards etc, the number of vertexes, edges and sides, comparing the length of edges, the shape and the size of the sides, understanding what opposite and neighboring sides are and their position. Naming and describing cubic objects: the number of vertexes, edges and sides, comparing the length of the edges, the shape and size of sides, understanding what opposite and neighboring sides are and their position.

Lesson 3: Giving a list of the characteristics of rectangular prisms and cubes by means of models. Making up various rectangular prisms using four matchboxes. Producing the reflections of the solid made from matchboxes. Finding objects in symmetrical arrangement in the classroom. Listing symmetrical objects. Defining the position of the planes of symmetry in case of various solids.

Lesson 4: Defining the position of the planes of symmetry in rectangular prisms and cubes. By using a model, learners studied the parallel and perpendicular

position of the opposite and the neighboring sides of rectangular solids, regular pentagon prisms, quadrilateral pyramids. Spreading rectangular prisms and cubes, examining the shape and size of the sides. Cutting squares from rectangles.

Lesson 5: The various grids of cubes. Studying the rectangles. The number of vertexes, opposite and neighboring vertexes, diagonal. Cutting the rectangle into two along the diagonal. Producing other plane figures by fitting the triangles gained in this way, and naming them. Gathering experience on plane figures and describing them. Further study of the rectangles: the number of sides, comparison of their length, determining the opposite and neighboring sides, the parallel position of the opposite sides, the perpendicular position of the neighboring sides. Measuring the sizes of angles by means of folded right angles.

Lesson 6: Studying the characteristics of plane figures made from two congruent right-angled triangles during the previous lesson: the number of vertexes and sides, defining the opposite and neighboring vertexes and sides, comparing the length of the sides, studying the parallel and perpendicular position of the opposite and the neighboring sides, the size of the angles. Comparing the characteristics of rectangles and parallelograms and highlighting their differences. Studying squares: the number of vertexes, opposite and neighboring vertexes, the diagonal. Cutting the square into two parts along the diagonal. Producing plane figures from the two right-angled isosceles triangles. Further study of squares: the number of sides, comparing their length, opposite and neighboring sides, the parallel and perpendicular position of the opposite and neighboring sides, the size of the angles.

Lesson 7: Demonstrating parallel and perpendicular pairs of straight lines as well as straight lines which are not parallel and perpendicular. Producing plane figures cut out from paper without restriction, and describing their characteristics. Listing the characteristics of rectangles and squares. Producing planes figures from the 2, 3, 4 and 6 regular triangles from the set of logics, which consists of 48 various plane figures, which can be red, yellow, blue or green. Their sizes are, small or large, their shape can be circle, square or triangle, their surface can be smooth or there is a hole in them. Making observations on parallel pairs of sides. Producing rectangles of different length and identical height from strips of paper.

Lesson 8: Producing various plane figures from paper strips by one cut. Naming them and describing their characteristics and shared characteristics. Cutting general rhombus from rectangle, its characteristics. Cutting general deltoid from rectangle, and its characteristics. Making rectangles and then the "frame" of a general parallelogram from six match sticks. Making squares then general rhombus from four match sticks. Comparing the characteristics of squares and rhombuses.

Lesson 9: Comparing the characteristics of squares and rectangles. Making 2 rectangles, a pentagon and a triangle, a triangle and a quadrangle, 2 quadrangles and 2 triangles from a rectangle by one cut. Drawing squares on square grid.

Lesson 10: Drawing various quadrangles on square grid. Drawing various triangles on grid. Drawing parallel and perpendicular pairs of straight lines.

Lesson 11: Coloring the parallel pairs of sides of the quadrangles drawn on grid and designating the right angles. In triangles coloring the sides perpendicular to each other. Drawing quadrangles according to given requirements. Studying the structure of the edges of rectangular prism and cubes. Observing the parallel and perpendicular edges.

Lesson 12: Producing reflection on plane through activity: folding a painted sheet of paper, on a black photographic paper folded into two making a pattern by running a pin through it, then unfolding it holding it in the direction of light. Cutting a given pattern from a sheet of paper folded into two parts. Observing reflections. On grid reflecting given figures on given axis. Producing figures symmetrical on axis by clipping.

Lesson 13: Finding the symmetry axes of plane figures cut out from paper by means of folding and mirror. Formulating experiences and observations. Drawing plane figures which have no symmetry axis, and which have exactly 1, 2, 3 and 4 symmetry axes.

Lesson 14: Producing figures symmetrical on axis on square grid. Selecting plane figures according to given characteristics. Formulating statements "every" and "there is such. . . ".

Lesson 15: Selecting plane figures according to given characteristics. Establishing the logical validity of statements. Drawing plane figures according to given requirements. Twenty questions.

Lesson 16: Post test.

References

[1] Ambrus, A., Bevezetés a matematikadidaktikába (Introduction to mathematics didactics), *Budapest, ELTE Eötvös Kiadó*, (in Hungarian) (1995).

[2] Árvainé Libor, I., Lángné Juhász, I., Szabados, A., Sokszínű matematika 3 (Colourful mathematics 3), *Szeged, Mozaik Kiadó*, (in Hungarian) (2005).

[3] Árvainé Libor, I., Lángné Juhász, I., Szabados, A., Sokszínű matematika 4 (Colourful mathematics 4), *Szeged, Mozaik Kiadó* (in Hungarian) (2006) p. 71.

[4] Bruner, J.S., Új utak az oktatás elméletéhez (Toward a Theory of Instruction), *Budapest, Gondolat* (in Hungarian) (1974).

[5] Neményi, E.C., Wéber, A., Matematika tankönyv általános iskola 3.osztály (Mathematics course books for grade 3 in primary schools), *Budapest, Nemzeti tankönyvkiadó*, (in Hungarian) (2005).

[6] Neményi, E.C., Káldi, É., Matematika tankönyv általános iskola 4. osztály (Mathematics course book for grade 4 in primary schools), *Budapest, Nemzeti tankönyvkiadó*, (in Hungarian) (2005).

[7] Dávid, L., A két Bolyai élete és munkássága (The life and ouvre of the two Bolyais), *Budapest, Gondolat*, (in Hungarian) (1979), p. 94.

[8] Holt, J., Iskolai kudarcok (Failures at school), *Budapest, Gondolat*, (in Hungarian) (1991), p. 99.

[9] Falus, I. (ed.), Bevezetés a pedagógiai kutatás módszereibe (Introduction to the methodology of pedagogical research), *Budapest, Műszaki Könyvkiadó*, (in Hungarian) (2000).

[10] Majoros, M., Oktassunk vagy buktassunk? (Shall we teach or fail the children?) *Budapest, Calibra* (in Hungarian) (1992).

[11] Peller, J., A matematikai ismeretszerzési folyamatról (On the acquisition of mathematical knowledge), *Budapest, ELTE Eötvös Kiadó*, (in Hungarian) (2003).

[12] Peller, J., A matematikai ismeretszerzés gyökerei (The roots of acquisition of mathematical knowledge), *Budapest, ELTE Eötvös Kiadó*, (in Hungarian) (2003).

[13] Piskalo, A.M., Geometria az 1–4. osztályban (Geometrics in classes 1–4), *Budapest, Tankönykiadó* (in Hungarian) (1977).

[14] Pólya, Gy., A gondolkodás iskolája (How to solve it), *Budapest, Akkord Kiadó* (2000), (in Hungarian) p. 197.

[15] Rakos, K., A mi matekunk 3. osztály (Our maths. Class 3), *Budapest, Nemzeti Tankönykiadó*, (2002), (in Hungarian) p. 149.

[16] Rakos, K., A mi matekunk 4. osztály (Our maths. Class 4), *Budapest, Nemzeti Tankönykiadó*, (in Hungarian) (2001).

[17] Revuz, A., Modern matematika – élő matematika (Modern mathematics – living mathematics), *Budapest, Gondolat*, (in Hungarian) (1973).

[18] Scherlein, M., Czakó, A., Hajdu, S., Lászlóné Novák, Matematika 3 (Mathematics 3), *Budapest, Műszaki Könyvkiadó*, (2005), (in Hungarian) p. 94.

[19] Scherlein, M., Czakó, A., Hajdu, S., Lászlóné Novák, Matematika 4 (Mathematics 4), *Budapest, Műszaki Könyvkiadó*, (2005), (in Hungarian).

[20] Skemp, R.R., A matematikatanulás pszichológiája (The psychology of learning mathematics), *Budapest, Gondolat Kiadó*, (1975), (in Hungarian) pp. 38–39.

[21] Szendrei, J., Gondolod, hogy egyre megy? (Do you think it is the same?) *Budapest, Typotex Kiadó*, (2005), (in Hungarian) p. 401.

[22] Tall, D., The Mathematical Growth, `http://davidtall.com`, (2004).

[23] Teppo, A., Van Hiele Levels of Geometric Thought Revisited, *Mathematics teacher*, March (1991).

[24] Török, T., Matematika II. általános iskola 3. osztály (Mathematics II. Primary school grade 3), *Budapest, Nemzeti Tankönykiadó*, (2002), (in Hungarian) p. 54.

[25] Török, T., Matematika II. általános iskola 4. osztály (Mathematics II. Primary school grade 4), *Budapest, Nemzeti Tankönykiadó*, (in Hungarian) (2002).

[26] Vigotszkij, L.S., Gondolkodás és beszéd (Cognition and speech), *Budapest, Trezor Kiadó*, (in Hungarian) (2000), p. 206.

Ibolya Szilágyiné Szinger
Eötvös József College, Baja, Hungary
e-mail: `szilagyine.szinger.ibolya@ejf.hu`

Annales Mathematicae et Informaticae
***35** (2008) pp. 189–204*
http://www.ektf.hu/ami

The usage of adapted ICT in the education of children with special educational need in different countries of Europe

Ildikó Tóthné Molnár[a], Tibor Radványi[b], Emőd Kovács[b]

[a]EGYMI, Budapest, Hungary

[b]EKF, Institute of Mathematics and Informatics, Eger, Hungary

Submitted 15 September 2008; Accepted 2 December 2008

Abstract

The education of the forthcoming generation is always a difficult task. This is particularly true for those educational institutes where students requiring special treatment are educated. These students are educationally challenged, mentally challenged or multi-challenged. In this article we present a Socrates-Comenius project which is dedicated to aim the usage of the information and communication technologies in the everyday educational work for students demanding special training. It is a fairly interesting and responsible challenge to discover how the computer could help to overcome difficulties and disadvantages which derive from handicap. On the other hand we present an international co-operation lasting for three years which prime goal was to construct a non-language-dependent software for handicapped children. In the article we introduce the institutions taking part in the development, the process of the program's creation, the steps to apply the program and the possibilities and methods of improvement.

Keywords: special educational, ICT, Technological tools, Manipulative materials

MSC: 68N01, 97U50, 97U60

1. Introduction

In the governing-principles of the European Union the use of information technologies in the public education is a highly supported area. The program of people living with disabilities and handicap has been appearing more and more emphatically in the educational program of the European Union. This program

enables the institutions of the participating countries to get financial support in order to proceed to work for their common desires and purposes. The generation of the planned project was inspired by the common interest and the mentioned governing-principles. To the partner Countries- The Netherlands, Germany, Latvia and Poland- joined Spain, Greece and Portugal as well. The common work started in 2004. The role of coordinator was accepted by the director of the Dutch institution.

1.1. The targets of the project

We set up as a common purpose to create a software for children requiring special education which is independent of any languages and cultures. The program is expected to:

- be able to adjust to national languages
- be adjustable to the type of handicap
- be improvable independently
- be applicable to any languages of any country
- that's why it should not demand the usage of it's written language in any conditions
- to adjust to the demands of the children
- It's content should be easily changeable and reedit able according to the required field of improvement.

It is also a part of the project to discover the partner institutions' level of knowledge in connection with informatics.

1.2. Introduction to concepts

"Handicap is the basic concept of the education of the handicapped. It marks that attribution or group of attributions which make a child be in need of education of the handicapped services." [1]

The new terminology calls it: specific educational demand. Mental deficiency/ handicap and special need: these two concepts spread in the educational and pedagogical word use. The difference between the two definitions is huge. Handicap refers to negative property, errors are assumed in the child. Special educational need is about demand. What do we mean by the phrase handicap?

Handicap is defined in various ways in Hungary as well:

- Mentally-handicapped individuals are those whose nervous system or any of their sense organs are injured and that's why their process of development differs from the development of healthy ones. [2]

- Handicap means a sequence of parameters which is usually extensive and defines the condition of an individual for a long time period.[3]

The categorization is based on the WHO system:

- The impairment is any kind of disorder in the psychological and physiological structure of a person.
- Handicap is altered and reduced capabilities of humans' certain activities.
- Detriment/disadvantage is the most serious, which derives from impairment or handicap and also means a kind of social disadvantage. So detriment is nothing else but the process when impairment and handicap become social factors. [4]

These days there was a significant change in the definition as the terminology defines it as special educational need. The emphasis was shifted to the need of special training, education, improvement and rehabilitation of a specific individual. The formation of this new view was prepared by many effects. The principle of normalization, the recognition of rights of handicapped children and the spread of inclusive school which evolved from the experience of integrated schooling. [5]

The European definitions slightly differ from the previous one. This is the ISCED system which was worked out by the UNESCO. The population of students who are affected by these problems can be subcategorized to three main categories:

- Those students whose learning difficulties compared to normal students have physiological origins. These are the different medical cases of organic disorders.
- Those students, whose learning difficulties are not possible to be properly explained, can't be directly joined to this factor.
- Those students, whose difficulties derive from different problems from their surroundings, namely socioeconomic, cultural/linguistic demands are not possible to be satisfied.

The following graph shows the ratio of students requiring special training in the population of school-aged students in the countries taking part in the project.

The autism is a so called pervasive development disorder, which covers three areas. The typical symptoms are the following: disturbance in establishing new relationships, language-communicational disorders, injury of those cognitive abilities which are in connection with the fantasy. Autism is defined as a so called spectrum disturbance because of the diversity of symptoms. Namely it's intensity can be quite various. For example: If we analyze mental abilities it can show mental handicap or high intelligence rate as well. The following can be read about the multi-handicapped children: It is a stage which comes into existence because of the effect of one or more biological injuries or impairments which can occur at the same time or independently from each other. It's result is a kind of defect

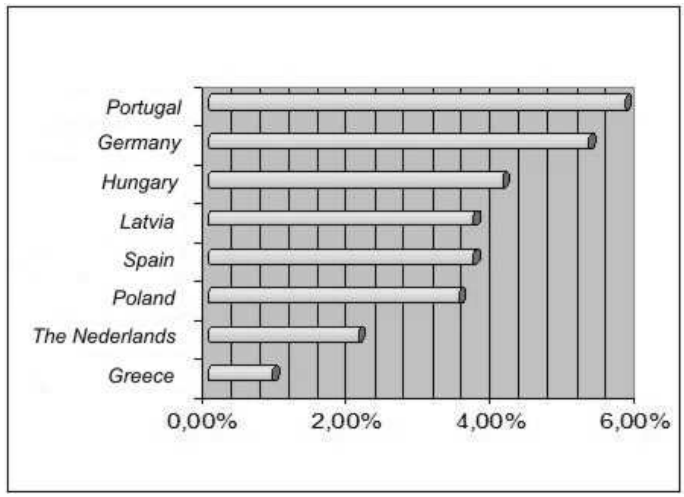

Figure 1: The population of school-aged students

expanding to several function areas. [5] The LXXIX. act in 1993 has set up educational obligations for the multi-handicapped people as well. During their education we have to concentrate in particular on their special needs, the reduction of their defenselessness and dependence, the development of the conditions of their communicational initiations. The capability of communication is the most important ability of humanity. It is difficult for a human to survive without it. By the help of this ability people create relationship with other people. The quality of this ability, and the impoundment of this ability affects a human's quality of life. [7] In the case of multi-handicapped children their evolutional disability makes their communication more difficult. That's why they need direct help and support to form their speech and to practice.

1.3. The introduction of the project

The participating countries in the project in Table 1:

In the project all of the eight institutions participated, each institution had one group of children with a membership of Six to eight with various ages. As one of our prime purposes was to develop the different areas of communication, so the participating children in the work were primarily multi-handicapped and autistic children. In both injuries the development of communication is a highly emphasized area. Unconventionally the program was not created only for them but they were the first users.

The Netherlands - Emiliusscool - Son en Breugel városa	Three to twenty year-old multi-handicapped children are educated.
Germany - Eberhart - Shomburg - Scule Latzen városa	Children between six and eighteen are educated.
Latvia - Berzupes Speciala Internatskola - Dobele városa	Children are between six and eighteen year old.
Poland - Szkola Podstawowa Specjaina - Gubin városa	From six to fourteen-year-old mentally handicapped children and autistic children are trained and educated.
Greece -Special School of Seress	Four to fourteen year-old children are accepted to attend the school.
Portugal - Pais e Amigos do Cidadao Deficiente Mental - Marihna Grande városa	The pupils study in class system between six and fifteen years meanwhile special support is given.
Spain - Frederico Gracia Lorca Centros - Madrid-Alcobendas	Accepting children from the age of three to the age of fourteen.
Hungary - Benedek Elek Óvoda, Általános Iskola, Speciális Szakiskola és EGYMI - Budapest	The education and upbringing of two hundred twenty-four mental-handicapped children is organized from the age of starting the nursery school to the age of entering special vocational schools.

Table 1: The participating countries in the project

1.4. The provision of appliances in the institutes

We decided to be a special part of the project to discover the level of knowledge in connection with informatics in the partner institutions. In order to reach this goal every institution filled in a questionnaire. The result of the survey from country to country.

Country	Result
The Netherlands	In the institution a computer lab operates with 10 computers. Moreover two computers are placed in every single classroom. One is used for educational purposes and the other one is for administration. Every computer has internet connection and web camera. In addition five notebooks are provided for the colleagues and ten for the children. In everyday education work projectors are used and they also have 3 digital boards.
Germany	In every single classroom there is a computer with a printer, which is used in everyday education. In addition they have four very well equipped computers which are used by the teachers for administration and preparation for the lessons. Three computers are situated in the corridors, which can be used during the breaks and after the lessons. All the computers have internet connection.
Latvia	The institution owns one computer room where ten computers can be found. These machines are applied for educational and administrative purposes as well. In every day work they do not use information technology appliances. In the institution only one computer has internet access.
Poland	The institution possesses five computers. Four out of them are used for educational purposes but exclusively for individual development work. One computer is for the colleagues to execute administrative tasks. They also have a digital camcorder and a digital camera.
Greece	Every classroom has computers but these are used only during the individual work. Moreover two highly equipped computers are available for the educators with internet connection to organize the administration. Other information technology appliances are not used during their everyday work.
Spain	In the institution a special classroom of computer studies operates with twelve computers. Moreover they use two computers in every classroom for educational purposes. For administration duties four computers are used. For their everyday work they can also use projector, digital camera, digital camcorder and web-camera as well. Every computer has internet access.

Country	Result
Portugal	They possess a special classroom, where three modern computers are ready to be used with internet access for conducting educational and administrational roles. In addition computers can be found in four classrooms. They have digital camera and digital camcorder as well.
Hungary	In the institution two highly equipped computer room can be found. Thirty-two computers are ready to be used for class educational purposes. The developer educators and speech therapists can also use three computers for their everyday work. Educators can use two computers for administration. For everyday educational roles, a projector, an overhead projector and two digitalboards are used. Every computer has internet access.

Table 2: The result of the survey from country to country

The questions were asked from the workers of the institutions and concentrated on the level of ICT knowledge, preparedness, their use of appliances and their expectations about the project and the partner institutes. The pedagogues, who take part in the project, were sorted according to their personal declaration about their qualification of informatics.

Country	Qualification of the pedagogues	Count
The Netherlands	Beginner user	15
	Intermediate user	80
	Professional user	10
	Renewing purpose(programmer)	5
Germany	Beginner user	6
	Intermediate user	9
	Professional user	3
	Renewing purpose(programmer)	1
Latvia	Beginner user	14
	Intermediate user	4
	Professional user	1
	Renewing purpose(programmer)	-
Poland	Beginner user	2
	Intermediate user	8
	Professional user	3
	Renewing purpose(programmer)	-
Greece	Beginner user	7
	Intermediate user	3
	Professional user	-
	Renewing purpose(programmer)	-
Spain	Beginner user	21
	Intermediate user	12
	Professional user	1
	Renewing purpose(programmer)	-
Portugal	Beginner user	4
	Intermediate user	8
	Professional user	1
	Renewing purpose(programmer)	-
Hungary	Beginner user	35
	Intermediate user	18
	Professional user	3
	Renewing purpose(programmer)	-

Table 3: The result of the survey from country to country

2. The use of ICT appliances in the education of the handicapped

The cognition of computer catered new perspectives for the education of the handicapped. An instrument/application got into our possession which is not specifically a mean of the education of the handicapped, but it is a fairly usable appliance in everyday work processes. This appliance possesses those properties which enable us to improve those abilities of our students which are missing or weak, without making them stigmatized. As the most crucial and complex point of the education of the handicapped is that the curriculum and therapy is optimized for children. That is why we have a unique opportunity in our hands with the use of computer.

The advantage of using informational and communicational techniques in the education is that it helps students to improve according to their own speed, it put an end to the sharp differences/boundaries among subjects and it also improves creativity. Adoption ICT in the life of an institution, which uses special curriculum, is crucially important. It enables injured and handicapped students to study in an easier way and more efficiently. If we assert that there isn't any child who is identical to another, so this is exponentially true for children with special educational need. The differentiated education-training is the only possible way for them to complete a successful career at school. Applying information technologies makes development work even more intensive.

Computer programs are excellently suitable for waking up the interest of handicapped children. Computer is an interactive appliance. If someone carries out an operation the computer reacts. All the humans like if there is a reaction after his/her work. This is crucially important in case of those people who got used to the fact that they can only carry out their activities if they rely on the help of other people. During our work one of the highlighted areas is the development of communication. Computer is not an aim but a mean. It doesn't substitute reality, but it helps to get to know, to discover reality in those cases when there are obstacles to discover it.

ICT can carry out numerous tasks in the fields of special education. IT can be used as:

- a mean of teaching
- a mean of studying
- a mean of communication
- an aid of therapy
- an aid of diagnosis

3. The presentation of the software

During the three-year-period from the autumn of 2004 to the autumn of 2007 8 special schools were working on creating and developing computer programs for students who suffer from learning problems and for handicapped students. These programs, which are created according to these principles, have to fulfill two requirements:

The software should not be too "childish", low-leveled.

The software can not be too difficult neither in it's content, nor in it's handling.

During the time period which was mentioned above the creators of the program were trying to find the clue of the following question: what kind of content would be attractive and inspiring for students learning in a special education school.

The development of communication skills is a long term task both for the autistic children and for the multi-handicapped children as well.

It is difficult for them to add meaning to cognition. Interpretation and understanding are limited. As verbal communication can be too abstract for them, we can help them with visual communicational means/tools to understand the connections between symbols and meanings better. During the process of our work we are making efforts to focus on these viewpoints.

In order to make the program possible to be used by handicapped children, the program offers two choices: we can choose between single-, and double-buttoned mouse use in the starting screen. The screen itself also functions automatically, thus we can choose between the two types of mouse use by clicking on the left mouse button at the appropriate time. In this way students can select type ,which is adequate for them, by themselves.

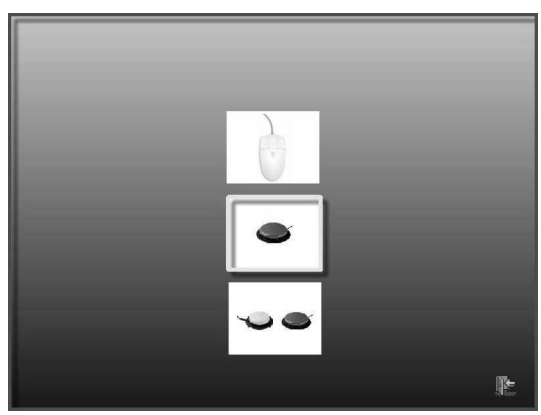

Figure 2: Select type

Other functions:

- By pressing any of the buttons the program returns to the main page
- Clicking on the speaker the program repeats the exercise.
- By Clicking on the "right arrow" we can step back to the exercise-choosing menu.
- If we click on the left-arrow we stay at the same level but we can choose again from the exercises.
- If we click on the "door" we can exit- when we click a new page appears where we have to confirm that we really want to exit.
- By clicking on the ALT and F4 buttons simultaneously we close the program. In the 8-picture version we can choose among four types of exercises.

Picture-book: memorizing task. The child can see the picture and hear the expression which belongs to it.

Reconciliation: comparative, conciliatory exercise, during this exercise one picture has to be chosen from the three minor ones which matches the major picture.

Lottery game: the child has to rely on the heard information to choose the suitable one from the pictures

Memory game: includes four different level conciliatory exercises.

Exercise types in the 24-picture version:

Picture-book: This is the learning phase, where we simultaneously see the picture and hear the sound which belongs to it. Listed under the adequate main concept.

Practice: This is also used for studying pictures and their names arranged to the adequate main concept. In the center a larger image can be seen, on which the main concept is situated. Around this larger image pictures can be found which belong to it. If we click on the picture we can hear their names again, meanwhile we see the enlarged picture.

Grouping: the student has to drag the picture into the suitable group.

Riddle: Which picture belongs to where? The pictures have to be taken into the correct group on the grounds of heard sound or observable picture.

In both formats a child gets feedback about his/her efficiency by using the same principle. In case of false or incorrect answer the student gets a short verbal instruction to try to solve the task again. In case of a correct answer the student gets a verbal affirmation or approval and at the end of the exercise a present is given, such as: a picture, sound or a short video is played.

Thus the format of the program includes 8- 8 and 24- 24 pictures and sounds belonging to them. To alter the inner content it is enough to change these with a simple copy operation. In this way teachers can easily prepare for their daily work, the help of a programmer is not needed.

The participating institutions in the project filled the program with their own inner content according to the field of improvement, the goals and interest of children. Some examples of the possibilities:

Social knowledge/Social science

Figure 3: Social knowledge

Fruits-vegetables

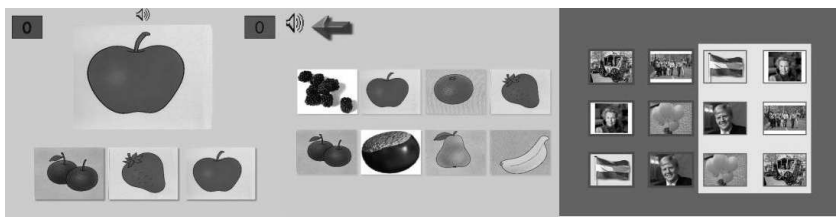

Figure 4: Fruits - vegetables

4. Altering the inner content of the program

Thus the format of the program includes 8–8 and 24–24 pictures and sounds belonging to them. To alter the inner content it is enough to change these with a simple copy operation.

Recording sounds. Sounds are possible to be recorded with the use of AUDACITY program:

- The program has already been copied from the CD, so we search for it in the folder
- We install the program by clicking on the SETUP icon, if it is needed, or:
- We start the program by using the headphones-like icon
- For use microphone is required

- Button with red spot → record
- Button with yellow square → stop
- Button with green arrow → play
- Double blue-lined button → pause
- scissors → cutting (the selected part is cut out from the recorded sound)
- IssI : those sounds are cut out which are not selected
- ssI-Iss → the selected part is cut out in a way that it is transformed to silence
- left and right arrows → the repeal of the final operation
- After the sound is ready it has to be saved

→ click on the FILE menu on the top of the left corner, and select EXPORT TO MP3 command from the appearing menu.

→ We give the name of the completed sound, and the location of the folder where we want to save. (`C:/Desktop/Rubricating/Sounds`)

Editing a picture

The edition of pictures is made possible by the use of the PAINT program.

- PAINT program can be found on the rest of the computers, on which Windows operation system is installed.
- Open START menu'select PROGRAMS/ALL PROGRAMS command' find ACCESSORIES folder'search for PAINT program and open PAINT by clicking on it.
- Buttons can be found on the left side of the program, with which we can draw in different ways, we can cut out, color a picture, resize it by using the PICTURE/IMAGE menu on the upper line of menus, and in addition we can also rotate pictures.

Renaming a picture

- Click on the picture, which is desired to be renamed, by the right button and choose the command of rename.
- At the name of the picture a word box appears, where we type the desired name, press ENTER, in this way the name changes.

Changing pictures and sounds

- Pictures are stored in the adequate folder, the pictures are renamed and we are assured that the format of the picture is JPEG.

- Sounds are also stored the adequate folder, after renaming and after being assured that the format of them is MP3.
- Open the folder which contains the pictures.
- From the top-line-menu select EDIT menu and click on it. From the appearing menu click on SELECT ALL command.
- Click on one of the selected pictures by using the right button of the mouse, then select COPY in the occurring menu.
- WE open the folder called: RUBRICATING/FŐFOGALOM ALÁ RENDEZÉS, DATA folder can be found here.
- Click on DATA folder with the right mouse button, and from the menu choose the INSERT command.
- At this point a question appears → Files with this name have already existed, overwrite (change them)? → this time we choose YES TO ALL command and the pictures get into the program.
- In case of sounds we use the same process. (Opening Sounds folder → edit menu → select all → right click → copy command → opening rubricating folder → right click on the DATA folder → insert → YES TO ALL button)

Creating word cards
The creation of word cards with the use of OP MAAT LOADER program:

- Instead of pictures we can also take word cards into the program, in this way we can get help to improve reading.
- The program can be found in the OP MAAT LOADER folder copied to the desktop, where we open the program by clicking on the icon which has the same name as the program.
- At the center of the appearing program window the CREATE CARD button can be found, by clicking on it a new window turns up where we can create word cards.
- We can type in the desired word to the word box which is under the blue rectangle (the more letters are typed in the less the size of the letters will be)
- Under it the PRINT TEXT button can be found, if we click on it the typed-in word appears on the blue word card.
- We have to save it with the appropriate name: → Click on the SAVE AS button under the blue rectangle and give the name of the picture and the location where we want to save.
- To place it into the program the instructions written above has to be followed.

5. Summary

This international version was created for the Tailor Made project initiated by Comenius. Wherever it is possible the format is independent from the text, in other words it can be adopted to any of languages in any countries. The uniqueness of this application is the following. While the structure of the program is kept it is easily possible to change the content of it according to the demands and requirements.

Our most important object was that the possibilities, which are provided by the software, should meet the requirements of the target group. The experience shows that other groups can also use the program efficiently not just those for whom the program was originally created.

It was not a difficult task for the participating schools to create a perfectly consistent and complete program from educational point of view, which can also be used for daily educational work. Besides we think it is very important that the program has to function as a wonderful entertainment and also as a teaching aid both for the students and for the teachers as well. The program can be applied according to the requirements of the students.

We can put it down for success that we got richer with a new method. A method which helps in preparing the arrangement of the theme/material which is adjustable to children. We could get to know each other's everyday life, and exchange each other's experience. The foreign-language knowledge and knowledge of informatics of the colleagues, who participate in the project, have improved significantly. These improvements promote the everyday use of information-technological applications in education.

The international co-operation provided insight to other nations' education in connection with children demanding special education. We could get to know how equipped are the institutions, concentrating on the quality of teaching and educational work we could also get acquainted to the information-communication technological application use.

During our work we discovered that Children used the created program easily and gladly. All the institutions presented the program during their own postgraduate courses. As after finishing the program there was no survey which would follow the further life of the program I can report only about observations in connection with our institution. The collaborating pedagogue colleagues participated in the familiarization with interest. They acquired the use and improvement of the program easily. They were experimenting with the alteration of the program with individual themes. All the colleagues use with everyday regularity the adopted version for improving individual abilities and skills and for class of logopedy. Currently we are planning to use it in the differentiated work of lessons.

References

[1] Zászkalicky P., Az egyéni deficittől a társadalmi felelősségig, *Új Pedagógiai Szemle* 2002.

[2] Csócsáné Horváth E., Kultúra – Fogyatékosság – Hátrányos helyzet, *Könyvtári Figyelő*, 1987. 5.sz. 528.p.

[3] Könczei Gy., Fogyatékosok a társadalomban, Budapest, *Gondolat*, 1992.13.p.

[4] Kálmán Zs., Könczei Gy., A Taigetosztól az esélyegyenlőségig, Budapest, *Osis*, 2002.

[5] Lányiné Engelmayer Á., Gyógypedagógiai Lexikon, Budapest, 2001.

[6] OECD 2000, Európai Iroda a Speciális Oktatás Fejlesztéséért, 2001.

[7] Hatos Gy., Az értelmi akadályozottsággal élő emberek: nevelésük, életük, 1996.

Ildikó Tóthné Molnár
Fővárosi Önkormányzat Benedek Elek Óvoda
Általános Iskola
Speciális Szakiskola
Magyarok Nagyasszonya tér 1–3.
H-1202 Budapest
Hungary

e-mail:
supernova69@citromail.hu

Tibor Radványi
Emőd Kovács
Eszterházy Károly College
Institute of Mathematics and Informatics
P.O. Box 43
H-3300 Eger
Hungary

e-mail:
radvanyi.tibor@ektf.hu
emod@ektf.hu