Обратимост на транслитерации

от Димитър Скордев

(декември 2005 г.)

1. Някои обратими версии на предложени транслитерации

В материала [10] се обсъждат два варианта на предложение за транслитерация от българска кирилица към латиница. Тези два варианта, които тук ще означаваме с α и β, могат да се опишат посредством следната таблица (при уславянето, че важи и онова, в което би се превърнала тя след повсеместна замяна на главните букви с малки, и че коя да е дума се транслитерира чрез замяна на нейните букви със съответните им низове):

	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ь	Ю	Я
α	A	B	V	G	D	E	Zh	Z	I	J	K	L	M	N	O	P	R	S	T	U	F	H	C	Ch	Sh	Sht	Y	J	Ju	Ja
β	A	B	V	G	D	E	X	Z	I	J	K	L	M	N	O	P	R	S	T	U	F	H	C	Q	W	Wt	Y	J	Ju	Ja

И при двата варианта не е налице обратимост поради следните причини: а) двете различни български букви **Й** и **Ь** се кодират с една и съща латинска; б) кодът на буквата **Ю** съвпада с резултата от транслитерацията на всеки от двубуквените низове **Йу** и **Ьу**, а кодът на буквата **Я** – с резултата от транслитерацията на всеки от двубуквените низове **Йа** и **Ьа**; в) кодът на буквата **Щ** съвпада с резултата от транслитерацията на двубуквения низ **Шт**. При варианта α обратимостта се нарушава още и поради това, че кодовете на буквите **Ж**, **Ч**, **Ш** и **Щ** съвпадат съответно с резултатите от транслитерацията на низовете **Зх**, **Цх**, **Сх** и **Схт**.

Една модификация на варианта β, при която е налице обратимост, е описана накратко в документа [8]. Модифицирания по този начин вариант β ще наричаме по-нататък вариант $\mathcal{D}$. Всъщност във варианта $\mathcal{D}$ се приема за българската азбука, че тя съдържа още една буква освен тридесетте, включени в горната таблица. Тази буква е наречена ударено **И**, но според лингвистите би било по-правилно да се нарича **И** с надреден знак с вид на ударение (вж. стр. 115 от [7]). Понеже тя се използва само при писането на някои местоимения, ще я наричаме за по-кратко *местоименно* **И**. Тук ще опишем друга обратима версия на варианта β, която ще означаваме с β' и която би могла в известен смисъл да се разглежда като леко опростена версия на варианта $\mathcal{D}$. Ще опишем и една обратима модифицирана версия α' на варианта α. Казано най-общо, при вариантите α' и β' се допуска в определени случаи кодът на една буква да съдържа допълнителен знак в началото си, който да сигнализира за отлика на конкретната употреба на въпросния код от някоя друга възможна негова употреба или да го отделя от кода на предходната буква. В качеството на такъв знак вземаме знака `/` (наклонена черта) и приемаме за код на местоименното **И** низа `/I` (пак при уславяне, че важи и онова, което би се получило чрез замяна на главните букви с малки). Заедно с това приемаме следните отклонения от кодирането при вариантите α и β, определено чрез таблицата, като точките 1–4 се отнасят както за варианта α', така и за варианта β', а точка 5 е само за варианта α' (разбира се приемаме също и всичко, което би се получило от споменатите няколко точки при замяна на главните букви с малки преди и след изразите "се кодира с" или "се кодират съответно с"), като освен това се уговаряме, щото всеки от евентуално наличните в подлежащия на транслитериране текст знаци `/` да се заменя с `//` при транслитерирането (би могло тази уговорка да се замени с по-сложна, според която знакът `/` се удвоява само в точно описан малък брой случаи, а в останалите се запазва без изменение):

1. Буквата **Й** се кодира с `/J` в случаите, когато е непосредствено след съгласна буква.
2. Буквата **Ь** се кодира с `/J` в случаите, когато не е непосредствено след съгласна буква.
3. Буквите **А** и **У** се кодират съответно с `/A` и `/U` в случаите, когато са непосредствено след някоя от буквите **Й** и **Ь** или от съответните им малки букви.

4. Буквата **Т** се кодира с `/T` в случаите, когато е непосредствено след буквата **Ш** или съответната ѝ малка буква.
5. Буквата **Х** се кодира с `/H` в случаите, когато е непосредствено след някоя от буквите **З**, **Ц**, **С** или от съответните им малки букви.

Пример 1. Таблицата по-долу показва как ще се транслитерират при вариантите α', β' и $\mathcal{D}$ няколко думи, които са с неоднозначно възстановяване по техните транслитерации поне при варианта α.

	изход	**Ицхак**	**Майа**	**пасха**	**Попйорданов**	**Таштепе**
α'	`iz/hod`	`Ic/hak`	`Maj/a`	`pas/ha`	`Pop/jordanov`	`Tash/tepe`
β'	`izhod`	`Ichak`	`Maj/a`	`pasha`	`Pop/jordanov`	`Taw/tepe`
$\mathcal{D}$	`izhod`	`Ichak`	`Maj|a`	`pasha`	`Popj#ordanov`	`Taw|tepe`

Вижда се, че транслитерациите при варианта β' без онази на името `Попйорданов`, могат да се получат от тези при варианта $\mathcal{D}$ чрез замяна на знака `|` със знака `/`. Това се дължи на факта, че при варианта $\mathcal{D}$ общо взето (само с няколко изключения[1]) важат точки 3 и 4, но със знака `|` вместо знака `/` (причината, поради която тук сме предпочели наклонената черта, е в това, че знакът `|` не принадлежи на основното множество от знаци, поддържани от GSM-апаратите(вж. [3]). По-сложно са решени обаче при варианта $\mathcal{D}$ проблемите по отношение на обратимостта, породени от еднаквото кодиране на **Й** и **Ь** (допълнително утежнени при този вариант от кодиране чрез същата латинска буква още и на местоименното **И**).

Забележка 1. Вариантите α' и β' остават обратими, ако ги допълним по следния начин с възможност за кодиране на коя да е ударена буква (като правим разлика между местоименно и ударено **И** в случаите, когато това е възможно и уместно): за да напишем кода на произволна такава буква, пишем първо кода, който би имала на това място буквата, ако беше без ударение, а след него поставяме низа `/'` – все едно, че коя да е ударена буква се тълкува като двубуквен низ, съставен от съответната неударена буква и друг знак с код `/'` (тълкуваме точките 1–5 по-горе като отнасящи се само за букви без ударения). Например буквата **ю** с ударение винаги ще има код `ju/'`, а буквата **у** с ударение ще бъде с код `/u/'`, когато е непосредствено след някоя от буквите **Й** и **Ь** или от съответните им малки букви, и ще има код `u/'` в останалите случаи (следователно, ако не е непосредствено след съгласна, двубуквеният низ, съставен от **й** и ударено **у**, ще се транслитерира като `j/u/'`, а чудноватият двубуквен низ, съставен от ударено **й** и ударено **у**, ще се транслитерира като `j/'u/'` в същия случай). Разбира се едва ли би имало пречки някаква подобна възможност за кодиране на ударените букви да се добави и към варианта $\mathcal{D}$.

Забележка 2. Вариантът $\mathcal{D}$ позволява обратимо транслитериране и на текстове, които могат да съдържат латински букви. За целта пасажите от текста, които са на латиница, се пренасят в транслитерирания текст, заградени с по един апостроф в началото и в края им, като обаче всички апострофи, които са били налице в оригиналния текст (независимо къде), се удвояват при транслитерирането. Няма пречки и във вариантите α' и β' да се предвиди същият начин на действие. Ще изложим обаче някои доводи в полза на известно негово модифициране, а именно, когато пренасяме в транслитерирания текст пасажите, които са на латиница, да се откажем от поставянето на апостроф веднага след тях и да постъпваме малко по-иначе, за да улесним транслитерирането. Да разгледаме например изречението

Замених `Windows` 98 `Second Edition` с `Windows` 2000.

Транслитерирано по начина, при който апострофите се поставят веднага след пасажите на латиница, това изречение добива вида

[1] В документа [8] не са изрично изброени изключенията от общите правила за кодиране на двойните букви и на съответните двойки единични. Може обаче да се приеме, че неявно или явно са посочени следните изключения: думите **вашта**, **вашто**, **нашта**, **нашто**, **пустошта** се транслитерират съответно чрез низовете `vawta`, `vawto`, `nawto`, `nawte`, `pustowta`, а пък лишените от смисъл думи **ваща**, **ващо**, **наща**, **нащо**, **пустоща** – съответно чрез `vaw^ta`, `vaw^to`, `naw^ta`, `naw^to`, `pustow^ta`.

```
Zamenih 'Windows 98 Second Edition' s 'Windows' 2000.
```

Технически удобно би било транслитерирането на кой да е текст да може се извършва чрез четене на текста отляво надясно знак по знак, като във всеки момент от работата да разполагаме с транслитерацията на вече прочетената част от текста и след прочитането на следващия знак (ако има такъв) да не се правят други промени в наличната вече транслитерация освен незабавно добавяне на такъв низ след нея, че пак да се получи транслитерацията на прочетеното до момента. Това обаче не може да стане в случая. И наистина, когато сме прочели частта "**Замених** `Windows`" от даденото изречение и сме получили частта "`Zamenih 'Windows`" от неговата транслитерация, няма как след прочитането на шпацията, която следва, само въз основа на прочетеното до момента да решим дали в транслитерацията трябва да добавим шпация или пък шпация, предхождана от апостроф. Яснота по този въпрос ще добием едва след прочитането на четирибуквения низ "**98** `S`" след въпросната шпация и значи преди този момент няма да имаме възможност да допълваме готовата част от транслитерацията. Подобно положение ще възникне и след прочитането на всяка от следващите шпации, които се предхождат от латинска буква. Например след прочитането на последната от тях ще получим възможност да допълним (и фактически да довършим) транслитерацията едва тогава, когато дочетем даденото изречение докрая.[2]

По-другият начин, за който споменахме, е следният: когато сме попаднали в латиница, ние ще поставим апостроф, оповестяващ връщане към кирилица, едва пред кода на първата след това прочетена буква на кирилица, когато има такава буква, и изобщо няма да поставим в противен случай. Ако работим така, то изречението, което разгледахме по-горе, би се транслитерирало в следния низ:

```
Zamenih 'Windows 98 Second Edition 's 'Windows 2000.
```

Този начин на работа води до не съвсем обичайно изглеждащи низове като горния, но пък има приятното свойство, че винаги, когато един текст е начало на даден текст, такова съотношение е налице и между транслитерациите на двата текста.

Забележка 3. Всеки от вариантите α' и β' може лесно да се допълни така, че да е приложим и за транслитерация от руска кирилица към латиница. Буквите от руската азбука, които липсват в българската, са три: **Ё**, **Ы** и **Э** (първата от тях нерядко се заменя с **Е**, но все пак е добре да се погрижим и за нейното кодиране). Една проста и сравнително естествена възможност, която може до известна степен да се мотивира с произношението на тези три букви, е те да се кодират съответно с `/O`, `/Y` и `/E` (други обратими начини за транслитерирация от руска кирилица към латиница са били предложени по-рано в [11, 12]). Всъщност за по-удобното транслитериране на текстове, в които преобладава руският език, би било уместно да се разменят кодовете на буквите **Ы** и **Ъ**, за да може по-често срещаната от двете да е с по-кратък код (разбира се въпросът за честотата на срещането им се отнася главно до съответните малки букви – случаите на употребата им като главни са много редки).

Забележка 4. Вместо да тръгваме от транслитерационна система, която не е обратима, а след това да я коригираме, бихме могли и директно да строим обратими транслитерационни системи. Такива са например системите α'' и β'', определени чрез таблицата по-долу при уславянето, направено във връзка с таблицата за системите α и β, и при уговорка за удвояване на знака `/` както във вариантите им α' и β'.

	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ь	Ю	Я
α''	A	B	V	G	D	E	/Z	Z	I	J	K	L	M	N	O	P	R	S	T	U	F	H	C	/C	/S	/T	Y	/J	/U	/A
β''	A	B	V	G	D	E	X	Z	I	J	K	L	M	N	O	P	R	S	T	U	F	H	C	Q	W	/T	Y	/J	/U	/A

Транслитерационната система β'' би могла да се пригоди за случая на руска кирилица с помощта на следните допълнения и изменения: буквите **Ё**, **Ы** и **Э** да се кодират съответно с `/O`, `/Y` и `/E`, а буквите **Щ** и **Ъ** – съответно с `/Q` и `/Y`.

[2] За някои текстове отлагането на допълването може да се окаже на места и значително по-голямо – например ако след някоя дума в текста, написана на латиница, следва дълга таблица, състояща се от числа.

2. Някои средства за доказване на обратимост

В определени случаи обратимостта на една транслитерация може да бъде доказана със средства, посочени в статията [6]. Към тези случаи спадат например транслитерациите, предложени в [11, 12], а също и транслитерациите α', β', α'' и β'' (включително и ако бъдат модифицирани по някой от начините, за които стана дума в забележки 1–4, стига от забележка 2 да бъде използван вторият от разгледаните там начини). Ще дадем кратко описание на споменатите средства.

Нека Σ и Δ са две азбуки, а T е изображение на Σ^* в множеството $\mathcal{P}(\Delta^*)$ на подмножествата на Δ^*, където Σ^* както обикновено се състои от всички крайни низове от символи на Σ (включително празния низ ε) и аналогично за Δ^*. Интуитивно можем да разглеждаме T като математическо описание на някоя система за транслитерация от Σ към Δ и за всеки низ ω от Σ^* да разглеждаме низовете, принадлежащи на множеството $T(\omega)$, като допустимите транслитерации на ω при тази система.[3] За всеки елемент ω на Σ^* елементите на $T(\omega)$ ще наричаме *образи* на ω относно T, а елемента ω ще наричаме *първообраз* относно T на всеки от тях. Изображението T ще наричаме *тотално*, ако всеки елемент на Σ^* има образ относно T, *сюрективно*, ако всеки елемент на Δ^* притежава първообраз относно T, *еднозначно*, ако никой елемент на Σ^* няма два различни образа относно T, и *обратимо*, ако никой елемент на Δ^* няма два различни първообраза относно T. *Обратното изображение* T^{-1} е изображението на Δ^* в $\mathcal{P}(\Sigma^*)$, дефинирано по следния начин: за всяко τ в Δ^* множеството $T^{-1}(\tau)$ се състои от първообразите на τ относно T. Очевидно T е тотално точно тогава, когато T^{-1} е сюрективно, и T е обратимо точно тогава, когато T^{-1} е еднозначно.

Ще казваме, че изображението T е *хомоморфно*, ако $T(\varepsilon) = \varepsilon$ и за всеки два низа ω_1 и ω_1 от Σ^* е в сила равенството $T(\omega_1\omega_2) = T(\omega_1)T(\omega_2)$ (дясната му страна означава множеството на всички конкатенации $\tau_1\tau_2$, където $\tau_1 \in T(\omega_1)$ и $\tau_2 \in T(\omega_2)$). Всяко изображение C на Σ в $\mathcal{P}(\Delta^*)$ може по единствен начин да се продължи до хомоморфно изображение на Σ^* в $\mathcal{P}(\Delta^*)$; за въпросното хомоморфно изображение ще казваме, че е *породено* от C. Интуитивно можем да разглеждаме елементите на $C(\sigma)$ за кой да е символ σ от Σ като допустимите негови кодове, а хомоморфното изображение, породено от C – като описание на транслитерацията, осъществявана чрез замяна на символите от Σ с допустими техни кодове.

Пример 2. Нека разликата $\Sigma \setminus \Delta$ се състои от всички български букви (без местоименното **И**), разликата $\Delta \setminus \Sigma$ – от всички букви на стандартната латиница, а сечението на Σ и Δ – от знака шпация, цифрите, препинателните знаци и евентуално други знаци, използвани както в българската, така и в английската писмена система. За всяка от транслитерационните системи α и β можем да разгледаме изображението c на Σ в Δ^*, което съпоставя на всяка българска буква нейния код, а на всеки от останалите символи от Σ – самия него. Нека C е изображението на Σ в $\mathcal{P}(\Delta^*)$, което на всеки символ σ от Σ съпоставя множеството с единствен елемент $c(\sigma)$. Тогава разглежданата транслитерационна система може да се представи чрез хомоморфното изображение T, породено от C. Това изображение е тотално и еднозначно, но не е обратимо. В случая на системата α то не е и сюрективно, но е такова в случая на системата β. И в двата случая изображението T^{-1} (което разбира се не е еднозначно) не е хомоморфно. Това е ясно например от обстоятелството, че $T^{-1}(\texttt{Ja})$ съдържа низ, който не може да се представи като конкатенация на низ от $T^{-1}(\texttt{J})$ и низ от $T^{-1}(\texttt{a})$.

Пример 3. Нека азбуките Σ и Δ са такива както в предходния пример, но с добавката, че тяхното сечение съдържа символа `/`. За всяка от транслитерационните системи α'' и β'' можем да разгледаме изображението c на Σ в Δ^*, което съпоставя на всяка българска буква нейния код, на символа `/` – низа `//`, а на всеки от останалите символи от Σ – самия него. Тогава разглежданата транслитерационна система може да се представи чрез хомоморфното

[3] При повечето реално използвани системи за транслитерации това множество ще бъде винаги от един елемент, но могат да се разглеждат и техни модификации, за които не винаги е така. Например би могло системата α да се модифицира по такъв начин, че еднобуквената дума **Я** да може да се транслитерира както чрез `Ja`, така и чрез `JA` (второто би било уместно например при нейно участие в по-дълъг текст, в който всички букви са главни). Или пък биха могли към системата α да се добавят някои неща от системата β, например да се разреши буквата **Ч** да може да се транслитерира както чрез `Ch`, така и чрез `Q`.

изображение T, дефинирано с помощта на изображението c по същия начин както в горния пример. Така дефинираното T ще бъде тотално, еднозначно и обратимо без да е сюрективно, като обратното му изображение пак няма да е хомоморфно.

Пример 4. Нека разликата $\Sigma\setminus\Delta$ се състои от всички руски букви (вкл. **Ё** и **ё**), разликата $\Delta\setminus\Sigma$ – от всички букви на стандартната латиница, а сечението на Σ и Δ – от знака шпация, цифрите, препинателните знаци и евентуално други знаци, използвани както в руската, така и в английската писмена система. Транслитерационната система, предложена в [12], може да се представи чрез хомоморфното изображение T, породено от изображението $\sigma\mapsto\{c(\sigma)\}$ на Σ в множеството на едноелементните подмножества на Δ^*, където $c(\sigma)=\sigma$ при $\sigma\in\Sigma\cap\Delta$, $c(\sigma)$ съвпада с кода на σ при транслитерацията α в случай че σ е руска буква, различна от буквите **Ё**, **Й**, **Х**, **Щ**, **Ъ**, **Ы**, **Э**, **Ю**, **Я** и от съответните им малки букви, като за току-що изброените девет главни руски букви съответствието c се определя чрез таблицата по-долу, а за съответните им малки букви – чрез таблицата, получаваща се от нея след замяна на главните букви с малки.

σ	**Ё**	**Й**	**Х**	**Щ**	**Ъ**	**Ы**	**Э**	**Ю**	**Я**
$c(\sigma)$	`Yo`	`Yj`	`Kh`	`Th`	`Jh`	`Ih`	`Eh`	`Yu`	`Ya`

И в този случай изображението T е тотално, еднозначно и обратимо, но не е сюрективно. Изображението T^{-1} отново не е хомоморфно.

Пример 5. При същите предположения за азбуките Σ и Δ както в пример 3 можем за транслитерацията β' по следния начин да дефинираме представящо я изображение T на Σ^* в множеството на едноелементните подмножества на Δ^*.[4] Нека c е изображение на Σ в Δ^*, което се отличава от едноименното изображение за транслитерацията β, използвано в пример 2, само по това, че преобразува символа `/` не в самия него, а в низа `//`. Да наречем една наредена двойка (ω,σ) от множеството $\Sigma^*\times\Sigma$ *особена*, ако е налице някой от следните случаи:

1. Низът ω завършва със съгласна буква, а символът σ принадлежи на множеството { **Й**, **й** }.
2. Низът ω не завършва със съгласна буква, а символът σ принадлежи на множеството { **Ь**, **ь** }.
3. Низът ω завършва с буква от множеството { **Й**, **й**, **Ь**, **ь** }, а символът σ принадлежи на множеството { **А**, **а**, **У**, **у** }.
4. Низът ω завършва с буква от множеството { **Ш**, **ш** }, а символът σ принадлежи на множеството { **Т**, **т** }.

Имайки на разположение това понятие, полагаме $T(\omega)=\{t(\omega)\}$, където изображението t на Σ^* в Δ^* се дефинира индуктивно така: полагаме $t(\varepsilon)=\varepsilon$, а за всеки низ ω от Σ^* и всеки символ σ от Σ приемаме, че $t(\omega\sigma)=t(\omega)c(\sigma)$, когато наредената двойка (ω,σ) не е особена, и $t(\omega\sigma)=t(\omega)\,/\,c(\sigma)$, когато тя е особена.[5] Така построеното изображение T е тотално, еднозначно и обратимо, но не е хомоморфно и не е сюрективно. Изображението T^{-1} също не е хомоморфно.

Нека Σ и Δ са две произволни азбуки, а T е хомоморфното изображение на Σ^* в $\mathcal{P}(\Delta^*)$, породено от дадено изображение C на Σ в $\mathcal{P}(\Delta^*)$. Очевидно T е тотално точно тогава, когато за всяко σ от Σ множеството $C(\sigma)$ има поне един елемент, а е еднозначно точно тогава, когато

[4] По аналогичен начин може да се дефинира такова изображение и за транслитерацията α'.

[5] Същото изображение t може да се дефинира и с помощта на един нормален алгоритъм (в смисъл на [9]), действащ в обединението на азбуките Σ и Δ, към което е добавен като спомагателен символът ♯. Може да се покаже, че за всеки низ ω от Σ^* низът $t(\omega)$ съвпада с резултата от прилагането към ω на нормалния алгоритъм, чиято схема изглежда така: в началото й са (независимо в какъв ред) всички формули за заместване от видовете $\delta\sharp\mathbf{Й}\to\delta\,\mathtt{J}\sharp$, $\delta\sharp\mathbf{й}\to\delta\,\mathtt{j}\sharp$, $\delta\sharp\mathbf{Ь}\to\delta\,\mathtt{J}\sharp$, $\delta\sharp\mathbf{ь}\to\delta\,\mathtt{j}\sharp$, където δ е някоя от латинските букви `B`, `C`, `D`, `F`, `G`, `H`, `K`, `L`, `M`, `N`, `P`, `Q`, `R`, `S`, `T`, `V`, `W`, `X`, `Z` и съответните им малки, както и формулите от вида $\delta\sharp\sigma\to\delta\,/\,c(\sigma)\sharp$, където $\delta\in\{$ `J`, `j` $\}$, $\sigma\in\{$ **А**, **а**, **У**, **у** $\}$ или $\delta\in\{$ `W` ,`w` $\}$, $\sigma\in\{$ **Т**, **т** $\}$, след това идват (пак независимо в какъв ред) всички формули за заместване от вида $\sharp\sigma\to c(\sigma)\sharp$, където $\sigma\in\Sigma\setminus\{$ **Ь**, **ь** $\}$, заедно с формулите $\sharp\mathbf{Ь}\to\mathtt{/J}\sharp$, $\sharp\mathbf{ь}\to\mathtt{/j}\sharp$ и накрая са (в този ред) формулите за заместване $\sharp\to\cdot$ и $\to\sharp$.

за всяко σ от Σ множеството $C(\sigma)$ има най-много един елемент. Лесно се проверява, че за да бъде изображението T обратимо, необходимо и достатъчно е да бъдат изпълнени следните две условия:

(a) $C(\sigma_1) \cap C(\sigma_2) = \emptyset$ за всеки два различни символа σ_1 и σ_2 от Σ.

(b) Никой низ от Δ^* не може да бъде представен по два различни начина като конкатенация на низове от обединението на всички множества $C(\sigma)$, съответни на символи σ от Σ.

В практически важния случай, когато множествата $C(\sigma)$ са крайни, проверката на условието (b) може винаги да се извърши посредством необходимото и достатъчно условие на Сардинас и Патерсън от статията [5]. В частност така може да се докаже обратимостта на изображенията T, разгледани в пример 3 и пример 4.

Възможността да се доказва обратимост чрез теоремата на Сардинас и Патерсън не прави безпредметно търсенето на други критерии за обратимост, а именно на такива, които дават само достатъчни условия за обратимост, но гарантират по-добро качество на обратимостта. Има поне две причини за това:

1. За удобството на една транслитерационна система е важна не само обратимостта, но и качеството на тази обратимост. Едно обратимо изображение T на Σ^* в $\mathcal{P}(\Delta^*)$ може да позволява за всяко ω от Σ^* лесно да се намери някой елемент на $T(\omega)$, но задачата за намиране на ω по даден елемент на $T(\omega)$ да бъде понякога по-трудна. Да разгледаме например транслитерационните системи α''^* и β''^*, които се получават от системите α'' и β'' от забележка 4 чрез размяна на местата на първия и втория символ във всички кодове, състоящи се от символа `/` и латинска буква след него (в частност буквата **Я** ще се кодира с `A/` вместо с `/A`). Системите α''^* и β''^* също са обратими, но е по-трудно (при четене отляво надясно) намирането на първообразите на някои низове, като да речем онези, в които първият символ е латинската буква `A` , а всички следващи са `/` – без да се прочете целият низ и да се установи дали дължината му е четна или нечетна няма да може да се открие дали първообразът му започва с **А** или с **Я** .
2. Измежду многобройните предлагани досега системи за транслитерация има такива, чиито съответни изображения T не са хомоморфни поради контекстна зависимост на кодирането на определени букви или поради използването на по-специални кодове на някои буквосъчетания. Такова е например положението със системите α' и β', в които кодирането на буквите **А**, **а**, **Й**, **й**, **Т**, **т**, **У**, **у**, **Ь**, **ь** (а също и на буквите **Х** и **х** в случая на α') зависи от контекста (контекстно зависимо кодиране се среща и в една от системите, предложени в [11], а очевидно и поставянето на апострофите при споменатото в първата част кодиране на смесени текстове, съдържащи и кирилица, и латиница, също е контекстно зависимо действие).

Споменатите по-горе други критерии ще се отнасят за изображения, осъществявани от последователни преобразователи в смисъла на § 3.3 от [2]. Според приетата там дефиниция *последователен преобразовател* с входна азбука Σ и изходна азбука Δ е наредена петорка от вида $(K, \Sigma, \Delta, H, s_0)$, където K е крайно множество (множество на *състоянията*), s_0 е елемент на K (*началното състояние*), H е крайно подмножество на декартовото произведение $K \times \Sigma^* \times \Delta^* \times K$ (елементите на H се наричат *ходове*).[6] Изображението T, осъществявано от един такъв последователен преобразовател се дефинира по следния начин: за всеки низ ω от Σ^* множеството $T(\omega)$ се състои от елементите τ на Δ^* със свойството, че за някое неотрицателно цяло число k, някои $\omega_1, \ldots, \omega_k$ от Σ^*, някои $\tau_1, \ldots, \tau_k$ от Δ^* и някои $s_1, \ldots, s_k$ от K четворките $(s_{i-1}, \omega_i, \tau_i, s_i)$, $i = 1, \ldots, k$, принадлежат на H и са в сила равенствата $\omega = \omega_1 \ldots \omega_k$, $\tau = \tau_1 \ldots \tau_k$.

Пример 6. Ако T е хомоморфно изображение на Σ^* в множеството на крайните подмножества на Δ^*, то T съвпада с преобразованието, осъществявано от последователния преобразовател $(K, \Sigma, \Delta, H, 0)$, в който $K = \{0\}$ и H се състои от всички наредени четворки $(0, \sigma, \theta, 0)$, където $\sigma \in \Sigma$ и $\theta \in T(\sigma)$.

[6] Тази терминология не е общоприета. Например в книгата [4] терминът последователен преобразовател означава нещо друго, като възприетото тук негово значение е по-близо до това на термина краен преобразовател в [4].

Пример 7. Като по-сложен пример за осъществяване на транслитерация от последователен преобразовател ще посочим осъществяването на изображението, съответно на транслитерацията β', от последователния преобразовател $(\{0,1,2,3\}, \Sigma, \Delta, H, 0)$, където азбуките Σ и Δ са същите както в примерите 3 и 5, а H се дефинира по следния начин: означаваме с c същото изображение както в пример 5 и за всяко $i \in \{0,1,2,3\}$ и всяко $\sigma \in \Sigma$ причисляваме към H наредената четворка (i, σ, τ, j), определена с помощта на равенствата

$$j = \begin{cases} 1, & \text{ако } \sigma \in \{\ \textbf{Й}, \textbf{й}, \textbf{Ь}, \textbf{ь}\ \}, \\ 2, & \text{ако } \sigma \text{ е съгласна буква, различна от } \textbf{Ш} \text{ и } \textbf{ш}, \\ 3, & \text{ако } \sigma \in \{\ \textbf{Ш}, \textbf{ш}\ \}, \\ 0 & \text{в останалите случаи,} \end{cases}$$

$\tau = / c(\sigma)$ при $i = 0$, $\sigma \in \{\ \textbf{Ь}, \textbf{ь}\ \}$, при $i = 1$, $\sigma \in \{\ \textbf{А}, \textbf{а}\ , \textbf{У}\ , \textbf{у}\ \}$, при $i = 2$, $\sigma \in \{\ \textbf{Й}, \textbf{й}\ \}$ и при $i = 3$, $\sigma \in \{\ \textbf{Й}, \textbf{й}\ , \textbf{Т}\ , \textbf{т}\ \}$, а във всички останали случаи $\tau = c(\sigma)$.[7]

Пример 8. Ще построим последователен преобразовател, който да осъществява изображението, съответно на транслитерацията β'' след допълването ѝ с възможност за транслитериране на смесени текстове по начина, описан в забележка 2 (аналогичен последователностен преобразовател може да се построи и във връзка с транслитерацията α''). Ще предполагаме, че азбуката Δ се състои от всички главни и малки букви на стандартната латиница, знака шпация, цифрите, препинателните знаци, символа / , знака апостроф и евентуално други знаци, използвани както в българската, така и в английската писмена система, а азбуката Σ се получава от Δ чрез добавяне на главните и малките български букви. Да означим с c изображението на Σ в Δ^*, което съпоставя на българските букви техните кодове при транслитерацията β'', удвоява символа / и знака апостроф и запазва без промяна всеки друг символ от Σ. Последователният преобразовател, за който стана дума, има вида $(\{0,1\}, \Sigma, \Delta, H, 0)$, като H се състои от всички наредени четворки $(0, \sigma, c(\sigma), 0)$, където σ е символ от Σ, но не е латинска буква, всички наредени четворки $(0, \sigma, \text{'}\sigma, 1)$, където σ е латинска буква, всички наредени четворки $(1, \sigma, c(\sigma), 1)$, където σ е символ от Σ, но не е българска буква, и всички наредени четворки $(1, \sigma, \text{'}c(\sigma), 0)$, където σ е българска буква,

Ако един последователен преобразовател осъществява дадено изображение T на Σ^* в $\mathcal{P}(\Delta^*)$, то обратният преобразовател осъществява изображението T^{-1} на Δ^* в $\mathcal{P}(\Sigma^*)$ (вж. задача 5 към § 3.3 на [2]). Това дава възможност за единообразно изследване на сложността на някои транслитерации и на техните обратни преобразования, а именно чрез изследване на сложността на изображението, осъществявано от произволен последователен преобразовател.

Достатъчните условия, които ще формулираме, ще използват няколко помощни понятия. Ще наричаме *последователен анализатор* всяка наредена четворка (K, Γ, G, s_0), където Γ е някоя азбука, K е крайно множество (множество на *състоянията*), s_0 е елемент на K (*началното състояние*), G (множеството на *ходовете*) е крайно подмножество на декартовото произведение $K \times \Gamma^* \times K$. За всяко t_0 в K ще наричаме *път на* (K, Γ, G, s_0) *с начало* t_0 всяка крайна редица

$$t_0, \psi_1, t_1, \psi_2, t_2, \ldots, t_{m-2}, \psi_{m-1}, t_{m-1}, \psi_m, t_m, \tag{1}$$

такава, че тройките (t_0, ψ_1, t_1), (t_1, ψ_2, t_2), $\ldots$, $(t_{m-2}, \psi_{m-1}, t_{m-1})$, (t_{m-1}, ψ_m, t_m) принадлежат на G (допуска се да имаме $m = 0$, т.е. редицата (1) да се състои само от t_0). *Резултат* от този път ще наричаме низа $\psi_1, \psi_2, \ldots, \psi_{m-1}, \psi_m$ (при $m = 0$ това ще бъде празният низ). Всеки път на (K, Γ, G, s_0) с начало s_0 ще наричаме *анализ* на резултата от пътя.

На всеки последователен преобразовател $(K, \Sigma, \Delta, H, s_0)$ съпоставяме два последователни анализатора – неговия *входен анализатор* (K, Σ, H_1, s_0) и неговия *изходен анализатор* (K, Σ, H_2, s_0), където H_1 и H_2 се състоят съответно от проекциите (k, ω, k') и (k, τ, k') на четворките (k, ω, τ, k'), принадлежащи на H. Очевидно е в сила следното твърдение, което дава възможност за използване на входни и изходни анализатори при въпросите за еднозначност и за обратимост на изображение, осъществявано от последователен преобразовател.

[7] С леко усложняване на този последователен преобразовател можем да получим и такъв, който пък да осъществява изображението, съответно на транслитерацията α'.

Основни достатъчни условия за еднозначност и обратимост. Нека T е изображението, осъществявано от даден последователен преобразовател $(K, \Sigma, \Delta, H, s_0)$. За да бъде изображението T еднозначно, достатъчно е да бъдат изпълнени следните две условия:

(i) не съществува низ от Σ^* с повече от един анализ чрез входния анализатор на преобразователя $(K, \Sigma, \Delta, H, s_0)$;

(ii) при всеки избор на k и k' в K и на ω в Σ^* съществува най-много един низ τ от Δ^*, за който четворката (k, ω, τ, k') принадлежи на H.

За да бъде изображението T обратимо, достатъчно е да бъдат изпълнени следните две условия:

(iii) не съществува низ от Δ^* с повече от един анализ чрез изходния анализатор на преобразователя $(K, \Sigma, \Delta, H, s_0)$;

(iv) при всеки избор на k и k' в K и на τ в Δ^* съществува най-много един низ ω от Σ^*, за който четворката (k, ω, τ, k') принадлежи на H.

Разбира се условието (i) е изпълнено за всеки последователен преобразовател от вида, посочен в пример 6, а условието (ii) в този случай е еквивалентно с изискването за всяко σ от Σ съответното множество $T(\sigma)$ да има най-много елемент (това изискване е и необходимо за еднозначността на хомоморфното изображение, осъществавано от преобразователя). Лесна е проверката на тези условия и за последователните преобразователи, разгледани в примерите 7 и 8.

За да улесним проверката на условията (iii) и (iv) поне в част от случаите, когато те са изпълнени, ще разгледаме един алгоритъм за търсене на анализи на низове при даден последователен анализатор. Въпросния алгоритъм ще наречем *LRLMS-алгоритъм* ("LRLMS" идва от "left-to-right longest-match strategy" – термин, близък до някои, срещащи се в работи по компютърна лингвистика като например [1]). Нека са дадени един последователен анализатор (K, Γ, G, s_0) и един низ θ от Γ^*. Да разгледаме следната частична операция върху анализи чрез (K, Γ, G, s_0) на същински начала на θ: ако

$$s_0, \varphi_1, s_1, \varphi_2, s_2, \ldots, s_{k-2}, \varphi_{i-1}, s_{i-1}, \varphi_i, s_i \tag{2}$$

е един такъв анализ, то търсим тройка (s_i, φ, s) от G, за която низът $\varphi_1\varphi_2 \ldots \varphi_{i-1}\varphi_i\varphi$ е начало на θ с максималната възможна дължина, и в случай, че съществува точно една такава тройка, добавяме нейните компоненти φ и s към редицата (2) като нови предпоследен и последен членове. С помощта на тази операция действието на LRLMS-алгоритъма върху низа θ при дадения последователен анализатор (K, Γ, G, s_0) може да се опише така: тръгваме от едноелементната редица с единствен член s_0 и прилагаме въпросната операция дотогава, докато се получи анализ на целия низ θ или пък, без да е получен такъв анализ, по-нататъшното прилагане на операцията се окаже невъзможно. Приемаме, че прилагането на алгориъма е успешно, ако след някакъв брой негови стъпки възникне първият от тези два случая.

Един анализ (2) чрез последователния анализатор (K, Γ, G, s_0) ще наричаме *LRLMS-съобразен*, ако този анализ може да се получи от своя резултат чрез успешно прилагане на LRLMS-алгоритъма. Това изискване е тривиално изпълнено при $i = 0$, а при $i > 0$ то е еквивалентно с това да не съществуват такъв номер j от множеството $\{1, \ldots, i\}$ и такава тройка (s_{j-1}, φ, k) от G, различна от $(s_{j-1}, \varphi_j, s_j)$, че φ е начало на низа $\varphi_j\varphi_{j+1} \ldots \varphi_{i-1}\varphi_i$ и φ_j е начало на φ.

Пример 9. Нека T е хомоморфно изображение, представящо транслитерационната система α (вж. пример 2), а (K, Γ, G, s_0) е изходният анализатор на съответния последователен преобразовател от пример 6. Тогава низът **pasha** има следните два анализа чрез споменатия анализатор: 0, p, 0, a, 0, s, 0, h, 0, a, 0 и 0, p, 0, a, 0, **sh**, 0, a. От тях само вторият е LRLMS-съобразен.

Пример 10. Нека T е хомоморфно изображение, представящо някоя от транслитерационните системи α''^* и β''^*(вж. т. 1 на стр. 6), а (K, Γ, G, s_0) е изходният анализатор на съответния последователен преобразовател от пример 6. Тогава низът A//B има точно един

анализ чрез споменатия анализатор, а именно 0, A, 0, //, 0, B, 0. Този анализ обаче не е LRLMS-съобразен.

Забележка 5. Ако в множеството G на даден последователен анализатор (K, Γ, G, s_0) няма тройки с празна втора компонента, то прилагането на LRLMS-алгоритъма към кой да е низ от Γ^*, който не притежава LRLMS-съобразен анализ, ще завърши неуспешно след някакъв брой стъпки. Например в условията на пример 10 работата на алгоритъма завършва неуспешно след достигането на пътя 0, A/, 0.

Ще наричаме един последователен анализатор (K, Γ, G, s_0) *LRLMS-съобразен*, ако всички анализи чрез него са LRLMS-съобразни. Разбира се от тази дефиниция следва, че ако анализаторът (K, Γ, G, s_0) е LRLMS-съобразен, то не може един низ от Γ^* да има два различни анализа чрез този анализатор.

Да наречем едно състояние на даден последователен анализатор *достижимо*, ако то е последен член на някой анализ чрез този анализатор.

Необходимо и достатъчно условие за LRLMS-съобразност на един последователен анализатор. Един последователен анализатор (K, Γ, G, s_0) е LRLMS-съобразен тогава и само тогава, когато са изпълнени следните условия:

(A) не съществува тройка от G с достижима първа компонента и празна втора;

(B) не съществуват две тройки от G с достижима първа компонента, които се различават само по третите си компоненти;

(C) за никоя тройка (t_0, φ, t) от G с достижимо t_0 не съществува такъв път (1) на (K, Γ, G, s_0) с $m > 1$, че $\psi_1\psi_2 \ldots \psi_{m-1}$ е същинско начало на φ, а φ е начало на $\psi_1\psi_2 \ldots \psi_{m-1}\psi_m$.

Следствие 1. Нека (K, Γ, G, s_0) е такъв последователен анализатор, че в G няма тройки с празна втора компонента и не съществуват две тройки от G, които да се различават само по третите си компоненти. Нека в G няма и такива тройки (t_0, φ, t), (t_0, ψ_1, t_1) и (t_1, ψ_2, t_2), че ψ_1 да е същинско начало на φ и някой от низовете $\psi_1\psi_2$ и φ да е начало на другия. Тогава преобразователят (K, Γ, G, s_0) е LRLMS-съобразен.

Следствие 2. Нека (K, Γ, G, s_0) е последователен анализатор, за който $K = \{s_0\}$, и нека W е множеството на вторите компоненти на тройките от G. Нека всички низове от W са непразни и не съществуват такива низове φ, ψ_1 и ψ_2 от W, че ψ_1 да е същинско начало на φ и някой от низовете $\psi_1\psi_2$ и φ да е начало на другия. Тогава преобразователят (K, Γ, G, s_0) е LRLMS-съобразен.

3. Някои конкретни приложения

Да предположим, че са дадени две азбуки Σ и Δ, а T е изображението на Σ^* в $\mathcal{P}(\Delta^*)$, което описва дадена транслитерационна система. Ще казваме, че тази система е *лесно използваема*, ако T е обратимо и може да бъде осъществено от някой последователен преобразовател с LRLMS-съобразни входен и изходен анализатор. Като вземем пред вид основните достатъчни условия за еднозначност и обратимост, виждаме, че току-що изказаното условие със сигурност е изпълнено, ако T се осъществява от някой последователен преобразовател $(K, \Sigma, \Delta, H, s_0)$, удовлетворяващ условието (iv), на който входният и изходният анализатори са LRLMS-съобразни. Оказва се, че такъв последователен преобразовател може да се намери за всяко от изображенията T, разгледани в примерите 3, 4, 5 и 8. За изображенията от примерите 3 и 4 може да се използва съответният последователен преобразовател, построен по начина, посочен в пример 6, за изображението от пример 5 – последователният преобразовател, построен в пример 7, а за изображението от пример 8 – последователният преобразовател, построен в същия този пример. За всеки от споменатите преобразователи проверката на условието (iv) е проста, затова ще се спрем само на въпроса за LRLMS-съобразността на техните входни и изходни анализатори.

При преобразователите, отговарящи на първите два от споменатите примери, LRLMS-съобразността на входния и изходния анализатор може да се докаже с помощта на следствие 2. За входните анализатори условието от следствието е изпълнено по простата причина,

че всички низове от съответното множество W са с дължина 1. За изходните анализатори в случая на пример 3 това условие е изпълнено, понеже пак никой низ от W не е същинско начало на низ от W. Малко по-сложно се проверява условието за изходния анализатор в случая на пример 4 – използва се, че когато един низ от W е същинско начало на друг, вторият се получава от първия чрез добавяне на буквата `h` , а от друга страна никой низ от W не започва с тази буква.

За преобразователите, построени в примерите 7 и 8, забелязваме, че в четворките, които са техни ходове, втората компонента е винаги с дължина 1, а освен това първата и втората компоненти определят еднозначно четвъртата (а също и третата, но това в случая не е съществено). Оттук е ясно, че за всеки от съответните входни анализатори е изпълнено условието от следствие 1. С малко повече грижи можем да установим също, че в тези четворки първата и третата компонента определят еднозначно втората, а следователно и четвъртата. Поради това за изходните анализатори на тези преобразователи остава само да се покаже невъзможността да имат такива ходове (t_0, φ, t), (t_0, ψ_1, t_1) и (t_1, ψ_2, t_2), щото ψ_1 да е същинско начало на φ и някой от низовете $\psi_1\psi_2$ и φ да е начало на другия. В случая на пример 7 можем да използваме, че ако бихме имали такива ходове на изходния анализатор, то ще трябва φ да е някой от низовете `Wt`, `wt`, `Ja`, `ja`, `Ju`, `ju`, а ψ_1 да бъде първата буква на този низ. Проверява се обаче, че тогава t_1 трябва да е 1 или 3, а ψ_2 да започва със символа `/` – нещо, което не е възможно. В случая на пример 8 пък още предположението, че ψ_1 е същинско начало на φ, води до противоречие при положение, че φ и ψ_1 са втори компоненти на ходове на изходния анализатор.

Литература

[1] Gerdemann, D., van Noord, G. Transducers from rewrite rules with backreferences. In: *Ninth Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics* (8–12 June 1999, Bergen, Norway). San Francisco, Morgan Kaufmann Publishers, 1999, 126–133. `http://acl.ldc.upenn.edu/E/E99/`

[2] Ginsburg, S. The Mathematical Theory of Context-Free Languages. McGraw–Hill, 1966.

[3] GSM 03.38 character set. `http://www.csoft.co.uk/sms/character_sets/gsm.htm`

[4] Lothaire, M. Algebraic Combinatorics on Words. Cambridge University Press, 2002.

[5] Sardinas, A., Patterson, C. A necessary and sufficient condition for the unique decomposition of coded messages. *IRE Intern. Conv. Record*, **8**, 1953, 104–108.

[6] Skordev, D. Transliteration and longest match strategy. *International Journal "Information Theories and Applications"* (под печат).

[7] БАН. Нов правописен речник на българския език. София, 2002.

[8] Добрев, Д. Предложение за транслитерация. `http://www.metodii.com/Transliteraciya.html`

[9] Марков, А. А. Теория алгорифмов. *Труды Математ. инст. им. В. А. Стеклова*, **42**, Москва-Ленинград, Изд. АН СССР, 1954.

[10] Скордев, Д. Някои предложения за транслитерация. `http://www.fmi.uni-sofia.bg/fmi/logic/skordev/transli0.htm`

[11] Успенский, В. А. К проблеме транслитерации русских текстов латинскими буквами. В: *Научно-техническая информация*, серия 2, *Информационные процессы и системы*. 1967, № 7, 12–20 (също в [13], 390–412).

[12] Успенский, В. А. Невто́н–Ньюто́н–Нью́тон, или Сколько сторон имеет языковой знак? В: *Русистика. Славистика. Индоевропеистика. Сборник к 60-летию Андрея Анатольевича Зализняка*. Москва, "Индрик", 1996, 598–659 (също в [13], 483–561).

[13] Успенский, В. А. Труды по нематематике. Москва, ОГИ, 2002. `http://www.mccme.ru/free-books/usp.htm`