The Unambiguous English-Russian Transliteration

Andrey Ponomarev
Moscow Institute of Physics and Technology

Abstract

Transliteration is writing a text in one language in the alphabet of another. Most of computer systems seen by the author operate in English, using Latin alphabet by default. Thus transliterating of Russian words on gadgets is rather common. Since there is no straight matching between English and Russian letters, the usual transliteration is often ambiguous.

The author proposes an algorithm of unambiguous English-Russian transliteration. Hopefully, the algorithm is simple enough to be used not only in computer programs but also in reading and writing by humans.

In order to construct the transliteration, a special mathematical concept of a system of rules is introduced and exploited. In some sense, it may resemble the classic grammar from the theory of formal languages. Two theorems about systems of rules are proven, which can also be interesting on their own.

Unambiguity is necessary when transliterating names, especially in computer systems. Via this, English and non-English software facilities can be compatible.

1 Introduction

One day the author wondered if it is possible to construct an unambiguous transliteration between the English and the Russian alphabets. The aim is to match English and Russian words strictly one-to-one so that the matching is as close as possible to the usual letter-wise transliteration. A word, of course, means a formal string here, that is, any finite sequence of letters.

The answer is yes, this is possible. But how much such a transliteration can meet our desires and how simple it can be are more complicated questions. In this article, a certain mathematical concept is proposed and one of countless possible one-to-one English-Russian transliterations is built, which, in the author's opinion, is both simple and linguistically convenient.

Where is such a transliteration needed? Most of computer systems known to the author operate in English, most of programming languages are based on the English language, the default set of characters in computers is Latin. When introducing the Russian language, it is important to maintain compatibility of names: for example, so that a program module with an English name could be called in Russian and vice versa. So unambiguity is essential in transliterating of names.

The author hopes that the article will be interesting even for readers not acquainted with the Russian language.

2 An alphabet. A language

For a start, let us recount the basic definitions and the notation to be used further.

Definition 1. *An alphabet* $\mathcal{X}$ is a nonempty finite set; the elements of an alphabet are called *symbols*, or *letters*. In prospect we will work with two specific alphabets: the English $\mathcal{A} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, ..., \mathrm{z}\}$ and the Russian

$$\mathcal{R} = \{\text{а, б, в, г, д, е, ё, ж, з, и, й, к, л, м, н, о, п, р, с, т, у, ф, х, ц, ч, ш, щ, ъ, ы, ь, э, ю, я}\}$$

Definition 2. *A string* on an alphabet $\mathcal{X}$ is a finite sequence of symbols from $\mathcal{X}$: $\omega = \widetilde{x}_1\widetilde{x}_2...\widetilde{x}_n$, $n \in \mathbb{N}_0$. The empty string is shown as ε. We denote strings by lower-case Greek letters.

Definition 3. Let $\omega_1 = \widetilde{x}_1\widetilde{x}_2...\widetilde{x}_n,\ \omega_2 = \widetilde{y}_1\widetilde{y}_2...\widetilde{y}_m$ be strings on $\mathcal{X}$. *The concatenation* of ω_1 and ω_2 is $\omega = \widetilde{x}_1\widetilde{x}_2...\widetilde{x}_n\widetilde{y}_1\widetilde{y}_2...\widetilde{y}_m$. We write it as $\omega = \omega_1\omega_2$.

Definition 4. Let ω_1 and ω_2 be strings. We call ω_2 *a substring* of ω_1 if there exist some strings α, β such that $\omega_1 = \alpha\omega_2\beta$.

Definition 5. *A language* on an alphabet $\mathcal{X}$ is any set of strings on $\mathcal{X}$. The set of *all* strings on $\mathcal{X}$ is shown as $\mathcal{X}^*$. The language $\mathcal{A}^*$ will be called English for short, the language $\mathcal{R}^*$ – Russian.

An unambiguous English-Russian transliteration is a bijective function $F : \mathcal{R}^* \longrightarrow \mathcal{A}^*$.

3 A partition of a string. A system of rules

Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary alphabets. We would like to have some instrument for convenient description of functions $F : \mathcal{X}^* \longrightarrow \mathcal{Y}^*$. Let us construct it.

Definition 6. *A rule* from $\mathcal{X}^*$ to $\mathcal{Y}^*$ is a pair (ξ, η) where ξ and η are some nonempty strings, $\xi \in \mathcal{X}^*,\ \eta \in \mathcal{Y}^*$. A rule can be represented more graphically as $\xi \longrightarrow \eta$. A rule is an «order to translation».

Next it is natural to consider a set of rules and to make it somehow generate a function $F : \mathcal{X}^* \longrightarrow \mathcal{Y}^*$.

Definition 7. Let ω be a string on an alphabet $\mathcal{X}$. *A partition* of ω into n pieces is a finite sequence $T = (t_0, t_1, ..., t_n),\ n \in \mathbb{N}$, where $t_0 = 0,\ t_n = |\omega|,\ t_0 < t_1 < ... < t_n$. Applying a partition to a string, we will get a sequence of n corresponding substrings: let $\omega = \widetilde{x}_1\widetilde{x}_2...\widetilde{x}_m$, then $T(\omega) = (\omega_1, \omega_2, ..., \omega_n)$, where $\omega_i = \widetilde{x}_{t_{i-1}+1}\widetilde{x}_{t_{i-1}+2}...\widetilde{x}_{t_i}$.

Example 1. Let $\omega =$ абвг, $T = (0, 1, 3, 4)$. Then $T(\omega) = ($а, бв, г$)$.

Definition 8. We say that of two partitions T_1 and T_2 of a string ω T_1 is greater than T_2 and write $T_1 > T_2$ if $T_1 \subsetneq T_2$, that is, T_1 is a strict subset of T_2. If $T_1 > T_2$ then T_2 can be made of T_1 by adding one or several «cuts» of ω.

Remark 1. The order on the set of all partitions of a string ω defined above is not linear: in general, there exist some incomparable partitions. For example, $T_1 = (0, 1, 3)$ and $T_2 = (0, 2, 3)$.

Definition 9. Let $\mathcal{T}$ be a set of partitions of a string ω, $T_0 \in \mathcal{T}$. T_0 is *a maximal element* of $\mathcal{T}$ if $\forall T \in \mathcal{T}\ (T \not> T_0)$, that is, there is no element greater than T_0. T_0 is *the greatest element* of $\mathcal{T}$ if $\forall T \in \mathcal{T}\ (T_0 > T)$.

Definition 10. Let $\Gamma = \{(\xi_1, \eta_1), (\xi_2, \eta_2), ...\}$ be a finite or countably infinite set of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$. A partition T of a string ω is called *acceptable* relatively to Γ if $\exists i_1, i_2, ..., i_n \in \mathbb{N} :$ $T(\omega) = (\xi_{i_1}, \xi_{i_2}, ..., \xi_{i_n}),\ (\xi_{i_1}, \eta_{i_1}), ..., (\xi_{i_n}, \eta_{i_n}) \in \Gamma$.

Definition 11. *A correct system of rules*, or simply *a system of rules*, from $\mathcal{X}^*$ to $\mathcal{Y}^*$ is a finite or countably infinite set of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$ $\Gamma = \{(\xi_1, \eta_1), (\xi_2, \eta_2), ...\}$ such that:
1) all the left parts ξ_i of the rules are different;
2) for any string $\omega \in \mathcal{X}^*$ in the set of its acceptable partitions relatively to Γ, there exists the greatest element.

Remark 2. The existence of the greatest element in the set of acceptable partitions includes that the set is not empty.

Example 2. Let $\mathcal{X} = \{$к, с$\} \subset \mathcal{R}$ («с» is Russian «s»), $\mathcal{Y} = \mathcal{A}$. Then $\Gamma = \{$к $\longrightarrow$ k, с $\longrightarrow$ $\longrightarrow$ s, кс $\longrightarrow$ x$\}$ is a correct system of rules. Indeed, if the substring кс enters in the string ω for m times then there is precisely 2^m acceptable partitions (each кс can be left whole or cut into (к, с)). There is, obviously, the greatest one among them – that one where all the кс substrings are whole.

Definition 12. Let $\omega \in \mathcal{X}^*$, Γ be a system of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$. There exists the greatest acceptable partition T of a string ω by Γ; $T(\omega) = (\xi_{i_1}, \xi_{i_2}, ..., \xi_{i_n})$, $(\xi_{i_1}, \eta_{i_1}), ..., (\xi_{i_n}, \eta_{i_n}) \in \Gamma$. *An image* of ω by the system of rules Γ is the string $\Gamma(\omega) = \eta_{i_1}\eta_{i_2}...\eta_{i_n}$, that is obtained by concatenation of the right parts of the corresponding rules.

Example 3. Consider the system of rules Γ from the example 2. $\Gamma(\varepsilon) = \varepsilon$ (the any acceptable partition is (0)), $\Gamma(\text{к}) = \text{k}$, $\Gamma(\text{ккссксксс}) = \text{kxsxx}$.

Remark 3. A system of rules Γ from $\mathcal{X}^*$ to $\mathcal{Y}^*$ defines a function $F : \mathcal{X}^* \longrightarrow \mathcal{Y}^*$, $F(\omega) = \Gamma(\omega)$, such that $F(\varepsilon) = \varepsilon$.

Proposition 1. *Let $F : \mathcal{X}^* \longrightarrow \mathcal{Y}^*$ be an arbitrary function from $\mathcal{X}^*$ to $\mathcal{Y}^*$ such that $F(\varepsilon) = \varepsilon$. Then there exists a system of rules Γ defining F.*

Proof. $\mathcal{X}^*$ is countably infinite; consider its enumeration $\mathcal{X}^* = \{\omega_i\}_{i\in\mathbb{N}}$, where all the ω_i are different. Let $\Gamma = \{(\omega_i, F(\omega_i))\}_{i\in\mathbb{N}}$. Γ is a correct system of rules and it denotes F. □

Proposition 2. *Let Γ be a correct system of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$. Then $\forall \widetilde{x} \in \mathcal{X}\ \exists \eta \in \mathcal{Y}^* :$ $(\widetilde{x}, \eta) \in \Gamma$.*

Proof. Consider the string $\omega = \widetilde{x}$ (of one symbol). There exists the greatest acceptable partition of ω relatively to Γ, but there is only one partition of ω – $T = (0, 1)$. So it is acceptable, therefore $\exists \eta \in \mathcal{Y}^* :\ (\widetilde{x}, \eta) \in \Gamma$. □

Theorem 1. *Let $\Gamma = \{(\xi_1, \eta_1), (\xi_2, \eta_2), ...\}$ be a finite or countably infinite set of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$. Then Γ is a correct system of rules $\Longleftrightarrow$ the following conditions are satisfied:*
1) all the left parts ξ_i of the rules are different;
2) $\forall \widetilde{x} \in \mathcal{X}\ \exists \eta \in \mathcal{Y}^* :\ (\widetilde{x}, \eta) \in \Gamma$;
3) $\forall i, j \in \mathbb{N}\ \forall \alpha\beta\gamma \in \mathcal{X}^*\ (\alpha, \beta, \gamma \neq \varepsilon,\ \xi_i = \alpha\beta,\ \xi_j = \beta\gamma \Longrightarrow \exists \eta \in \mathcal{Y}^* :\ (\alpha\beta\gamma, \eta) \in \Gamma)$.
The third item means that if some two left parts ξ_i and ξ_j of the rules «intersect» with the end of the first and the beginning of the second then there is a rule with the resultant «overlay» in its left part.

Proof. ($\Longrightarrow$) Item 1 derives from the definition; item 2 – from proposition 2. Let us establish item 3. Let $(\xi_i, \eta_i), (\xi_j, \eta_j) \in \Gamma$, $\alpha, \beta, \gamma \neq \varepsilon$, $\xi_i = \alpha\beta$, $\xi_j = \beta\gamma$. Let $\omega = \alpha\beta\gamma$. $T_1 = (0, |\alpha| + |\beta|, |\alpha| + |\beta| + 1, ..., |\alpha| + |\beta| + |\gamma|)$ and $T_2 = (0, 1, ..., |\alpha|, |\alpha| + |\beta| + |\gamma|)$ are acceptable partitions of the string ω relatively to Γ, because in Γ there are the rules $(\alpha\beta, \eta_i), (\beta\gamma, \eta_j)$ and, according to proposition 2, there is a rule with a left part of one symbol for each symbol in $\mathcal{X}$. By the definition of a system of rules, for ω there exists the greatest acceptable partition T, $T > T_1$, $T > T_2$. That means $T \subset T_1$, $T \subset T_2 \Longrightarrow T \subset T_1 \cap T_2 = (0, |\alpha| + |\beta| + |\gamma|) \Longrightarrow T = (0, |\alpha| + |\beta| + |\gamma|)$. Thus in Γ there is a rule with the left part $\alpha\beta\gamma$.
($\Longleftarrow$) By contradiction: let Γ not be a correct system of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$ – so $\exists \omega \in \mathcal{X}^*$ not having the greatest acceptable partition. Let $\mathcal{T}$ be the set of all acceptable partitions of the string ω. It is not empty (item 1 implies that $(0, 1, ..., |\omega|) \in \mathcal{T}$), it is finite and does not have the greatest element $\Longrightarrow \exists T, U \in \mathcal{T} :\ T \not> U,\ U \not> T$ and $\forall V \in \mathcal{T}\ (V \not> T$ and $V \not> U)$. T and U are maximal elements of $\mathcal{T}$ and are incomparable with each other. Let $T = (t_0, t_1, ..., t_n)$, $U = (u_0, u_1, ..., u_m)$. Let an index k be such that $\forall i \in \overline{0, k}\ (t_i = u_i)$, but $t_{k+1} \neq u_{k+1}$. Without loss of generality $t_{k+1} < u_{k+1}$. Let an index p be such that $\forall i \in \overline{0, p-1}\ (t_i < u_{k+1})$, but $t_p \geq u_{k+1}$. Obviously, $p > k + 1$. If $t_p = u_{k+1}$ then the partition $(t_0, t_1, ..., t_k(= u_k), t_p(= u_{k+1}), t_{p+1}, ..., t_n)$ is acceptable and also greater than T – a contradiction with the choice of T. So $t_p > u_{k+1}$. $\omega = \widetilde{x}_1\widetilde{x}_2...\widetilde{x}_{|\omega|}$, $\widetilde{x}_j \in \mathcal{X}$. $\alpha = \widetilde{x}_{t_k+1}...\widetilde{x}_{t_{p-1}}$, $\beta = \widetilde{x}_{t_{p-1}+1}...\widetilde{x}_{u_{k+1}}$, $\gamma = \widetilde{x}_{u_{k+1}+1}...\widetilde{x}_{t_p}$. $\alpha\beta$ is the part number $k + 1$ in the partition U; $\beta\gamma$ is the part number p in the partition T. By item 3 there exists a rule with the left part $\alpha\beta\gamma$, therefore the partition $(t_0, t_1, ..., t_k(= u_k), t_p, ..., t_n)$ is acceptable. But it is greater that T – a contradiction with the choice of T. □

Remark 4. The theorem above describes correct systems of rules more clearly than the original definition. It is easy to recognize with it if a certain set of rules is a correct system or not: it is enough to check the three conditions on the left parts ξ_i.

4 An algorithm generating the image by a system of rules

The aim of the intricate definition of a correct system of rules was to get freedom in translation of a string on one alphabet to a string on another.

Definition 13. Let ω be a string on an alphabet $\mathcal{X}$. *A segment* of the string ω is a pair $(a,b),\ a,b \in \mathbb{N}_0,\ a \le b \le |\omega|$, which we write as $[a,b]$ for convenience. Let $\omega = \widetilde{x}_1\widetilde{x}_2...\widetilde{x}_n,\ \widetilde{x}_i \in \mathcal{X}$. The substring of a segment $[a,b]$ is $\omega[a,b] = \widetilde{x}_{a+1}\widetilde{x}_{a+2}...\widetilde{x}_b$. In particular, $\omega[a,a] = \varepsilon$.

Definition 14. Let $[a,b]$ and $[c,d]$ be segments of a string ω. We say that $[a,b]$ is *greater* than $[c,d]$ and write $[a,b] > [c,d]$ if $a \le c,\ d \le b$ with $a < c$ or $d < b$ (that is, $[a,b] \ne [c,d]$).

Remark 5. As with the relation $>$ on the set of partitions of a string ω, the relation of order defined above is not linear, that is, generally there are incomparable segments. *The least* and *the greatest*, *a minimal* and *a maximal* elements are defined likewise.

Definition 15. Let $[a,b]$ be a segment of ω; $T = (t_0, t_1, ..., t_n)$ is a partition of ω. We say that T *includes* $[a,b]$ if $\exists i \in \mathbb{N}:\ t_{i-1} = a,\ t_i = b$.

Remark 6. $T(\omega) = (\omega[t_0,t_1], \omega[t_1,t_2], ..., \omega[t_{n-1},t_n])$.

Proposition 3. *Let $\Gamma = \{(\xi_1,\eta_1),(\xi_2,\eta_2),...\}$ be a system of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$; ω is a string on $\mathcal{X}$. Let $\exists i \in \mathbb{N}\ \exists c,d \in \mathbb{N}_0:\ \omega[c,d] = \xi_i$. Then in the greatest partition T of the string ω relatively to Γ, there is a segment $[a,b]$ such that $[a,b] \ge [c,d]$.*

Proof. According to proposition 2, $\forall \widetilde{x} \in \mathcal{X}\ \exists \eta \in \mathcal{Y}^*:\ (\widetilde{x},\eta) \in \Gamma$. Therefore the partition $U = (0,1,...,c,d,d+1,...,|\omega|)$ of the string ω is acceptable by Γ. $T > U$; that is, $T \subset U \Longrightarrow$ $\Longrightarrow \forall t \in \mathbb{N}_0\ (c < t < d \Longrightarrow t \notin T) \Longrightarrow \exists i \in \mathbb{N}:\ t_{i-1} \le c,\ t_i \ge d$. T includes the segment $[t_{i-1},t_i] \ge [c,d]$. □

Proposition 4. *Let $\Gamma = \{(\xi_1,\eta_1),(\xi_2,\eta_2),...\}$ be a system of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$; ω is a string on $\mathcal{X}$. Let $\mathcal{S}$ be the set of all the segments $[c,d]$ of ω such that $\exists i \in \mathbb{N}:\ \omega[c,d] = \xi_i$. Let $[a,b]$ be a maximal element of $\mathcal{S}$. Then $[a,b]$ is included into the greatest acceptable partition T of the string ω relatively to Γ.*

Proof. By proposition 3, there exists a segment $[a',b']$ of the string ω included into T such that $[a',b'] \ge [a,b]$. But $[a',b'] \in \mathcal{S}$ and in $\mathcal{S}$ there is no element greater than $[a,b] \Longrightarrow [a',b'] = [a,b]$, T includes $[a,b]$. □

Proposition 5. *Under the conditions of proposition 4, $[a,b]$ is a maximal element of $\mathcal{S}$, $(\xi_i,\eta_i) \in$ $\in \Gamma,\ \omega[a,b] = \xi_i$. We denote $\alpha = \omega[0,a],\ \beta = \omega[b,|\omega|]$; $\omega = \alpha\,\omega[a,b]\,\beta$. Then $\Gamma(\omega) =$ $= \Gamma(\alpha)\,\eta_i\,\Gamma(\beta)$.*

Proof. Let $T = (t_0,t_1,...,t_n)$ be the greatest acceptable partition of ω by Γ, $T(\omega) = (\xi_{j_1},\xi_{j_2},...,$ $\xi_{j_n})$. By proposition 4 T includes $[a,b]$, that is, $\exists k \in \mathbb{N}:\ a = t_{k-1},\ b = t_k,\ \xi_{j_k} = \xi_i,\ j_k = i$. Note that $U = (t_0,t_1,...,t_{k-1})$ and $V = (0, t_{k+1}-t_k, ..., t_n - t_k)$ are the greatest acceptable partitions of α and β, respectively (otherwise it is possible to construct a partition for ω greater than T). We have $\Gamma(\omega) = \eta_{j_1}...\eta_{j_{k-1}}\eta_i\eta_{j_{k+1}}...\eta_{j_n}$, $\Gamma(\alpha) = \eta_{j_1}...\eta_{j_{k-1}}$, $\Gamma(\beta) = \eta_{j_{k+1}}...\eta_{j_n}$. □

Algorithm 1 (of translation by a system of rules)**.** Let Γ be a system of rules from the language $\mathcal{X}^*$ to the language $\mathcal{Y}^*$; $\omega \in \mathcal{X}^*$. $\Gamma(\omega)$ is to be constructed.

1. If $\omega = \varepsilon$ than the result is found: $\Gamma(\omega) = \varepsilon$.
2. Let $\mathcal{S}$ be the set of segments $[c,d]$ of ω such that $\exists i \in \mathbb{N}:\ \omega[c,d] = \xi_i,\ (\xi_i,\eta_i) \in \Gamma$. Consider $[a,b]$ – a maximal element of $\mathcal{S}$.
3. Translate the strings $\alpha = \omega[0,a]$ and $\beta = \omega[b,|\omega|]$ by the system of rules Γ recursively.
4. The result is obtained by concatenation: $\Gamma(\omega) = \Gamma(\alpha)\eta_i\Gamma(\beta)$.

In step 2, a maximal element $[a,b]$ always exists because $\mathcal{S}$ is not empty and is finite. The algorithm is correct according to proposition 5.

Remark 7. The way of choosing $[a,b]$ is not defined in algorithm 1; there can be several maximal segments. One of the ways is to consider only segments of the form $[0,d]$ in the set $\mathcal{S}$, then $a=0,\ \alpha=\varepsilon,\ b=\max\limits_{[0,d]\in\mathcal{S}} d$. Such a choice is unambiguous; in this case, the recursive translation of step 3 corresponds moving over the string ω from the left to the right.

5 Systems of rules and bijective functions

Proposition 6 (generally known)**.** *Let A,B be arbitrary sets. Function $F:A\longrightarrow B$ is bijective $\Longleftrightarrow \exists F^{-1}:B\longrightarrow A$ such that $\forall a\in A\ (F^{-1}(F(a))=a)$ and $\forall b\in B\ (F(F^{-1}(b))=b)$. In this case F^{-1} is called the opposite function.*

Definition 16. Let $\Gamma=\{(\xi_1,\eta_1),(\xi_2,\eta_2),...\}$ be a finite or countably infinite set of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$. *The opposite set of rules* is $\Gamma^{-1}=\{(\eta_1,\xi_1),(\eta_2,\xi_2),...\}$.

We would like to describe a bijection $F:\mathcal{X}^*\longrightarrow\mathcal{Y}^*$ with a system of rules Γ so that Γ^{-1} would be a system of rules and denote F^{-1}. But there are some issues on the way to that.

Example 4. Consider the system of rules Γ from example 2. Γ^{-1} is not a system of rules from $\mathcal{Y}^*=\mathcal{A}^*$ to $\mathcal{X}^*=\{\text{к},\text{с}\}^*$.

Example 5. Consider Γ from example 2, but now replace $\mathcal{Y}$ with $\{\text{k},\text{s},\text{x}\}$. Now $\Gamma^{-1}=\{\text{k}\longrightarrow\text{к},\ \text{s}\longrightarrow\text{с},\ \text{x}\longrightarrow\text{кс}\}$ is a correct system of rules. However $F(\xi)=\Gamma(\xi),\ \xi\in\mathcal{X}^*$, is not a bijection (it is an injection but not a surjection: for example, the string ks does not have a prototype). So $G(\eta)=\Gamma^{-1}(\eta),\ \eta\in\mathcal{Y}^*$, cannot be the opposite to F.

Example 6. Consider a case when a system of rules Γ defines a bijective function $F:\mathcal{X}^*\longrightarrow\mathcal{Y}^*$, but Γ^{-1} is not a correct system of rules. Let $\mathcal{X}=\{0,1\},\ \mathcal{Y}=\{\text{y}\}$. We define an auxiliary function G as follows: $\omega\in\mathcal{X}^*$, then 1ω is a binary notation of a number $n\in\mathbb{N}$; let $\Gamma(\omega)$ be a string of $n-1$ symbols y. $G:\mathcal{X}^*\longrightarrow\mathcal{Y}^*$ is a bijection. Let $F(01)=G(00)=\text{yyy},\ F(00)=G(01)=\text{yyyy}$ and on all other arguments F equals G. Thus F is bijective. It is denoted by the system of rules $\Gamma=\{0\longrightarrow\text{y},\ 1\longrightarrow\text{yy},\ 00\longrightarrow\text{yyyy},\ 10\longrightarrow\text{yyyyy},\ 11\longrightarrow\text{yyyyyy}\}\cup\{\omega\longrightarrow F(\omega):\ |\omega|\geq 3\}$ (it contains rules for all ω except of 01, the image of which is the concatenation of the right parts of rules for 0 and 1). At that, Γ^{-1} is not a system of rules.

Theorem 2. *Let Γ be a system of rules from $\mathcal{X}^*$ to $\mathcal{Y}^*$ such that Γ^{-1} is a system of rules from $\mathcal{Y}^*$ ot $\mathcal{X}^*$. Let $F(\xi)=\Gamma(\xi),\ \xi\in\mathcal{X}^*$. Then F is a bijection $\Longleftrightarrow\Gamma^{-1}$ defines the opposite function F^{-1}.*

Proof. $\Longleftarrow$ Obvious by proposition 6.
$\Longrightarrow$ By contradiction: let $\exists\eta\in\mathcal{Y}^*:\ F^{-1}(\eta)\neq\Gamma^{-1}(\eta)$. We denote $\xi=F^{-1}(\eta),\ \omega=\Gamma^{-1}(\eta)$. $F(\xi)=\Gamma(\xi)=\eta$. By the definition of a system of rules there is the greatest acceptable partition T of ξ relatively to Γ; $\exists i_1,i_2,...,i_n\in\mathbb{N}:\ T(\xi)=(\xi_{i_1},\xi_{i_2},...,\xi_{i_n}),\ (\xi_{i_1},\eta_{i_1}),...,(\xi_{i_n},\eta_{i_n})\in\Gamma;\ \eta=\eta_{i_1}\eta_{i_2}...\eta_{i_n}$. Via this we get an acceptable partition U of the string η by Γ^{-1}: $U=(0,|\eta_{i_1}|,|\eta_{i_1}|+|\eta_{i_2}|,...,|\eta|)$; $U(\eta)=(\eta_{i_1},\eta_{i_2},...,\eta_{i_n})$. There exists the greatest partition V of η relatively to Γ^{-1}. $V\neq U$, otherwise $\Gamma^{-1}(\eta)=\xi$. So $V>U$. Let $U=(u_0,u_1,...,u_n),\ V=(v_0,v_1,...,v_m)$. In terms of sets $V\subset U\Longrightarrow\exists j_1,j_2\in\mathbb{N}_0\ \exists k\in\mathbb{N}:\ u_{j_1}=v_{k-1},\ u_{j_2}=v_k,\ j_2>j_1+1$, that is, in V there are two consecutive elements v_{k-1},v_k equal to elements u_{j_1},u_{j_2} of U, which are not consecutive. $\eta=\widetilde{y}_1\widetilde{y}_2...\widetilde{y}_{|\eta|},\ \widetilde{y}_j\in\mathcal{Y}$; we consider $\beta=\widetilde{y}_{u_{j_1}+1}\widetilde{y}_{u_{j_1}+2}...\widetilde{y}_{u_{j_2}}$. $\xi=\widetilde{x}_1\widetilde{x}_2...\widetilde{x}_{|\eta|},\ \widetilde{x}_j\in\mathcal{X}$; $T=(t_0,t_1,...,t_n)$; we consider $\alpha=\widetilde{x}_{t_{j_1}+1}\widetilde{x}_{t_{j_1}+2}...\widetilde{x}_{t_{j_2}}$. $T'=(0,t_{j_1+1}-t_{j_1},t_{j_1+2}-t_{j_1},...,t_{j_2}-t_{j_1})$ is an acceptable partition of α by Γ. Moreover, it is the greatest acceptable for α, because otherwise it would be possible to construct an acceptable partition of ξ greater than T. $T'(\alpha)=(\xi_{i_{j_1+1}},\xi_{i_{j_1+2}},...,\xi_{i_{j_2}})\Longrightarrow\Gamma(\alpha)=\eta_{i_{j_1+1}}\eta_{i_{j_1+2}}...\eta_{i_{j_2}}=\beta$. At that, β is the part number k in the partition V of the string $\eta\Longrightarrow\exists\gamma\in\mathcal{X}^*:\ (\beta,\gamma)\in\Gamma^{-1}$. So $(\gamma,\beta)\in\Gamma,\ F(\gamma)=\Gamma(\gamma)=\beta$. $\alpha\neq\gamma$, because in that case, for α there would be the acceptable partition $(0,|\alpha|)$, and by construction of α in the greatest acceptable partition T' of it there are at least three elements ($j_2>j_1+1$). Therefore $\alpha\neq\gamma,\ F(\alpha)=F(\gamma)=\beta$ – a contradiction with F being bijective. $\square$

6 Construction of the bijective English-Russian transliteration

In this section the previously built concepts are applied to construct a bijecive function from all the strings on the Russian alphabet to all the strings on the English one.

It is necessary to construct a bijective function $F : \mathcal{R}^* \longrightarrow \mathcal{A}^*$. We will describe it with a system of rules Γ such that Γ^{-1} is also a system of rules. Then, according to theorem 2, Γ^{-1} will denote the opposite function F^{-1}.

We shorten the considered alphabets: let $\mathcal{R}' \subset \mathcal{R}$, $\mathcal{A}' \subset \mathcal{A}$. Step by step we will add new letters and enrich Γ so that Γ and Γ^{-1} are systems of rules on the considered alphabets and so that the function defined by Γ is bijective. $\mathcal{R}'$ and $\mathcal{A}'$ will not be specified explicitly throughout the reasoning.

The choice of images for symbols sometimes does not have any clear logical explanation. Choosing them, the author followed two principles. First, the system of rules has to be as simple as possible for convenience of use. Second, the author followed their own linguistic intuition.

Since both Γ and Γ^{-1} are to be correct systems of rules, we write $\longleftrightarrow$ instead of $\longrightarrow$ in the rules for convenience.

Step 1 (letter-to-letter rules)**.**

а $\longleftrightarrow$ a	з $\longleftrightarrow$ z	н $\longleftrightarrow$ n	ф $\longleftrightarrow$ f
б $\longleftrightarrow$ b	и $\longleftrightarrow$ i	о $\longleftrightarrow$ o	х $\longleftrightarrow$ h
в $\longleftrightarrow$ v	й $\longleftrightarrow$ y	п $\longleftrightarrow$ p	ц $\longleftrightarrow$ c
г $\longleftrightarrow$ g	к $\longleftrightarrow$ k	р $\longleftrightarrow$ r	
д $\longleftrightarrow$ d	л $\longleftrightarrow$ l	с $\longleftrightarrow$ s	
ж $\longleftrightarrow$ j	м $\longleftrightarrow$ m	т $\longleftrightarrow$ t	

Step 2 (ч, ш, щ; an infinite chain of rules)**.**

Consider the following case. We would like to add the rule ш $\longleftrightarrow$ sh. But then Γ is no longer a bijection: Γ(ш) $=$ Γ(сх) $=$ sh. We have just «taken away» an image from any Russian string including сх as a substring – so we have to add a new rule for сх. Let it be сх $\longleftrightarrow$ skh. An issue of the same kind emerges: Γ(сх) $=$ Γ(скх) $=$ skh. It can be solved in the same way again and so on infinitely.

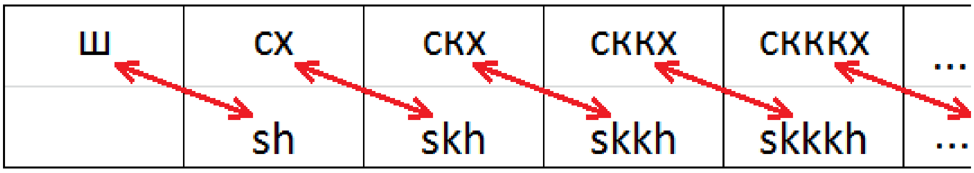

Figure 1: the chain of corrections for ш

Any finite number of corrections described does not solve the problem, but an infinity of them does (figure 1)! The redefinition of the images is achieved by adding a countably infinite chain of rules: ш $\longleftrightarrow$ sh, сх $\longleftrightarrow$ skh, скх $\longleftrightarrow$ skkh, сккх $\longleftrightarrow$ skkkh, ...

To denote such chains we introduce the following notation.

Definition 17. We establish the symbols $+$ and $*$. Let $\mathcal{X}$ and $\mathcal{Y}$ be alphabets; $n \in \mathbb{N}$; $\alpha_0, \alpha_1, ..., \alpha_n \in \mathcal{X}^*$, $\beta_0, \beta_1, ..., \beta_n \in \mathcal{Y}^*$ are some strings; $\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_n \in \mathcal{X}$, $\widetilde{y_1}, \widetilde{y_2}, ..., \widetilde{y_n} \in \mathcal{Y}$ are some symbols; $a_1, a_2, ..., a_n, b_1, b_2, ..., b_n \in \{+, *\}$. The expression

$$\alpha_0 \widetilde{x}_1^{\,a_1} \alpha_1 \widetilde{x}_2^{\,a_2} ... \widetilde{x}_n^{\,a_n} \alpha_n \longleftrightarrow \beta_0 \widetilde{y}_1^{\,b_1} \beta_1 \widetilde{y}_2^{\,b_2} ... \widetilde{y}_n^{\,b_n} \beta_n$$

represents the set of those and only those rules which have the form

$$\alpha_0 \widetilde{x}_1 ... \widetilde{x}_1 (i_1 \text{ times}) \alpha_1 \widetilde{x}_2 ... \widetilde{x}_2 (i_2 \text{ times}) ... \widetilde{x}_n ... \widetilde{x}_n (i_n \text{ times}) \alpha_n \longleftrightarrow$$
$$\longleftrightarrow \beta_0 \widetilde{y}_1 ... \widetilde{y}_1 (j_1 \text{ times}) \beta_1 \widetilde{y}_2 ... \widetilde{y}_2 (j_2 \text{ times}) ... \widetilde{y}_n ... \widetilde{y}_n (j_n \text{ times}) \beta_n$$

where the numbers i_k, j_k are related in each pair as follows:

1) if $a_k = *$ and $b_k = *$ then $0 \le i_k = j_k$;
2) if $a_k = *$ and $b_k = +$ then $0 \le i_k = j_k - 1$;
3) if $a_k = +$ and $b_k = *$ then $0 \le i_k - 1 = j_k$;
4) if $a_k = +$ and $b_k = +$ then $0 \le i_k - 1 = j_k - 1$.

We call this representation *the (*,+)-notation.*

Using the (*,+)-notation we can write the chain of rules for ш as ш $\longleftrightarrow$ sh, ск*х $\longleftrightarrow$ sk$^+$h. We call the symbols in the middle (from the both alphabets) *the delimiters.*

A corresponding chain of rules is constructed for ч: ч $\longleftrightarrow$ ch, цк*х $\longleftrightarrow$ ck$^+$h.

For щ we would like to add the rule щ $\longleftrightarrow$ shch. Then the image will be «lost» by all Russian strings including шч as a substring (note that exactly шч, not, for example, схцх, because the rules for ш and ч have already been introduced). The problem can be solved by the known approach, using the letter т/t as a delimiter: шт*ч $\longleftrightarrow$ sht$^+$ch.

So the following rules have been added:

ч $\longleftrightarrow$ ch	ш $\longleftrightarrow$ sh	щ $\longleftrightarrow$ shch
цк*х $\longleftrightarrow$ ck$^+$h	ск*х $\longleftrightarrow$ sk$^+$h	шт*ч $\longleftrightarrow$ sht$^+$ch

Step 3 (th и ph).

So far Γ^{-1}(th) = тх, which poorly represents the substance of this English letter combination. The author reckons as a good idea to introduce the rule тф $\longleftrightarrow$ th.

Note that now Γ(тх) = Γ(тф) = th and tf does not have a Russian prototype: the image of тх was «given» to тф. The simplest solution is to «give» тх the old image of тф, that is, to add the rule тх $\longleftrightarrow$ tf.

The translation Γ^{-1}(ph) = пх is also poor; Γ^{-1}(ph) = пф is much better. Proceed likewise!

So the following rules have been added:

тф $\longleftrightarrow$ th	тх $\longleftrightarrow$ tf	пф $\longleftrightarrow$ ph	пх $\longleftrightarrow$ pf

Step 4 (у, э and iotized vowels).

Let us add the rules у $\longleftrightarrow$ ou, ю $\longleftrightarrow$ u. This matching is suggested to the author by linguistic intuition.

We introduce the rules э $\longleftrightarrow$ oe, е $\longleftrightarrow$ e. The matching е $\longleftrightarrow$ e is exceedingly convenient and the rule for э is constructed similarly to the rule for у.

For ё and я we add the rules ё $\longleftrightarrow$ yo, я $\longleftrightarrow$ ya.

As in the step 2, in order to keep Γ bijective, we have to add infinite chains of rules with delimiter symbols. Let й be the delimiter for у and э, и – for ё and я. As a result we have

ю $\longleftrightarrow$ u		е $\longleftrightarrow$ e	
у $\longleftrightarrow$ ou	э $\longleftrightarrow$ oe	ё $\longleftrightarrow$ yo	я $\longleftrightarrow$ ya
ой*ю $\longleftrightarrow$ oy$^+$u	ой*е $\longleftrightarrow$ oy$^+$e	йи*о $\longleftrightarrow$ yi$^+$o	йи*а $\longleftrightarrow$ yi$^+$a

Remark 8. Currently Γ is not a correct system of rules: for example, there are rules with the left parts йо and ою, but no rule with йою (a condition of criterion 1 is not satisfied). We say that the chains for ё and у *collide.* This issue is to be resolved later.

Step 5 (ъ, ы, ь).

Having mustered all the present linguistic intuition and ideas about utility and aesthetics, the author has composed these rules: ъ $\longleftrightarrow$ oa, ы $\longleftrightarrow$ ea, ь $\longleftrightarrow$ ie.

Adding chains with the delimiters й and х (before а), we have

ъ $\longleftrightarrow$ oa	ы $\longleftrightarrow$ ea	ь $\longleftrightarrow$ ie
ох*а $\longleftrightarrow$ oh$^+$a	ех*а $\longleftrightarrow$ eh$^+$a	ий*е $\longleftrightarrow$ iy$^+$e

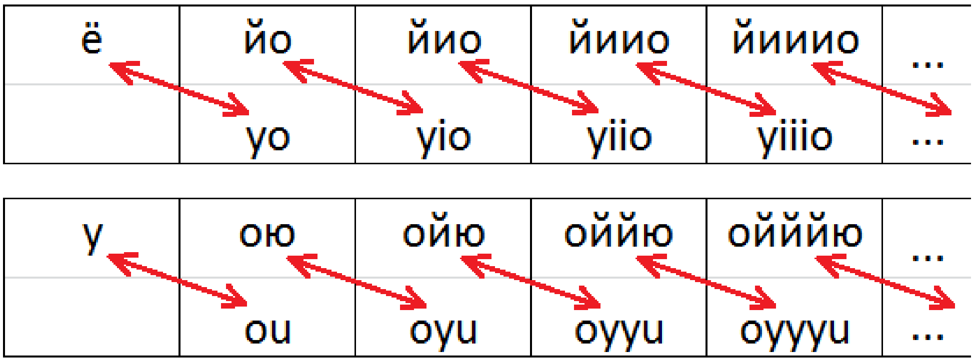

Figure 2: the chains of rules for ё and у

Step 6 (resolution of the vowel collisions: ё and у).

Consider the infinite chains of rules ё $\longleftrightarrow$ yo, йи*о $\longleftrightarrow$ yi$^+$o and у $\longleftrightarrow$ ou, ой*ю $\longleftrightarrow$ oy$^+$u.

As can be seen in figure 2, the chains are obtained by a «shift» of the matching by one position relatively to the letter-wise one. A rule for the new symbol is added to the beginning. We call this symbol *a head* of the chain.

Any string (except of the head) of the first chain and any string (except of the head) of the second chain in the same language can be «intersected» by one symbol (for example, йио and ою, yo and ou). Thus neither Γ nor Γ^{-1} is a system of rules unless the rules for all the «embracing» strings (as, for example, йиою и you) are added. This is *the collision*. In the considered case both of the heads are Russian, they belong to the same language – we call the collision *unilateral*.

For those strings which include the strings from the chains as substrings «without an intersection» the images are already defined correctly. Let us list all possible enters «with intersections» into an infinite table (figure 3). The horizontal index of a cell is the quantity of the first delimiter symbol, the vertical index – of the second one.

	йу	йиу	...
ёю	йою you	йиою yiou	...
ёйю	йойю yoyu	йиойю yioyu	...
⋮	⋮	⋮	

Figure 3: the table of «overlays»

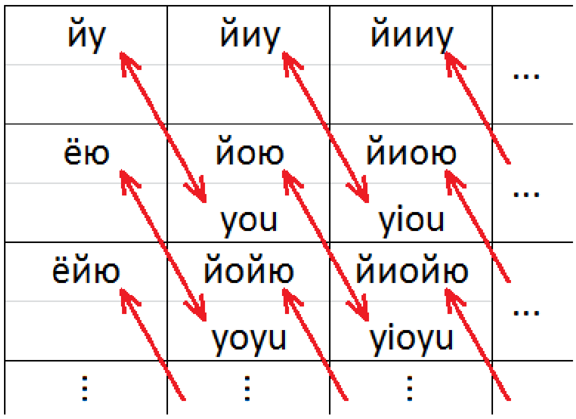

Figure 4: resolution of the collision

Russian strings are filling the whole table except of the top left cell, English strings – except of the first row and the first column. We need to pair all them. Let us shift the top row one cell to the left and match every Russian string with the English one from the bottom right of it (figure 4). All the obtained rules are added to Γ. Thus the entire countable infinity of the overlays is resolved.

The natural structure of the set of added rules is not one-dimensional (a chain) but two-dimensional. It can be represented in the (*,+)-notation:

йи*у $\longleftrightarrow$ yi*ou
ёй*ю $\longleftrightarrow$ yoy$^+$u
йи*ой*ю $\longleftrightarrow$ yi$^+$oy$^+$u

It is much easier to comprehend it as the following pattern: in the described case a delimiter is not required before a use of a rule with a head symbol. Examples:

1) Γ(йу) = you, Γ(йиу) = yiou, Γ(йииииу) = yiiiou: a rule with the head у is used, the delimiter is not required before it, that is, there is the same number of letters и as of letters i;
2) Γ(ёю) = yoyu, Γ(ёйю) = yoyyu, Γ(ёййййю) = yoyyyyu: a rule with the head ё is used, but a delimiter is required after it, that is, there is one letter y more than letters й;
3) Γ(йою) = yioyu, Γ(йиоййю) = yiioyyyu: no rules with a head symbol are used, all the delimiters are needed.

We are to return to this pattern later.

Step 7 (resolution of the vowel collisions: double collisions)**.**

The unilateral collision for the letters ё and у has just been resolved. After adding the rules for vowels, ъ, and ь five similar collisions emerged: ё-у, ё-ъ, ё-э, э-ы, ь-ы. They can be represented on a graph (figure 5) where vertices are English vowels and directed edges are two-letter images of Russian vowels, ъ, and ь.

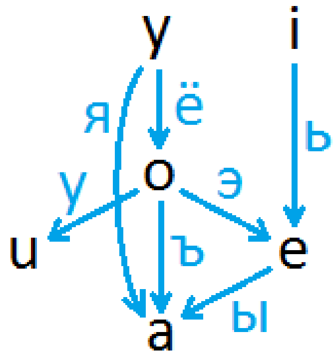

Figure 5: the vowels graph

A way of two edges in the graph corresponds to a collision of two letters, there is exactly five of them. There is also one way of three edges – ё-э-ы; it corresponds to a triple unilateral collision. There is no other ways (so as cycles) in the graph of vowels.

We solve the four remaining double collisions completely analogically to the collision ё-у, considered in the previous step. The same principle is true: a delimiter is not required before a use of a rule with a head symbol. The result can be represented in the (*,+)-notation:

$$\text{йи}^*\text{ъ} \longleftrightarrow \text{yi}^*\text{oa}$$
$$\text{ёх}^*\text{а} \longleftrightarrow \text{yoh}^+\text{a}$$
$$\text{йи}^*\text{ох}^*\text{а} \longleftrightarrow \text{yi}^+\text{oh}^+\text{a}$$

$$\text{ой}^*\text{ы} \longleftrightarrow \text{oy}^*\text{ea}$$
$$\text{эх}^*\text{а} \longleftrightarrow \text{oeh}^+\text{a}$$
$$\text{ой}^*\text{ех}^*\text{а} \longleftrightarrow \text{oy}^+\text{eh}^+\text{a}$$

$$\text{йи}^*\text{э} \longleftrightarrow \text{yi}^*\text{oe}$$
$$\text{ёй}^*\text{е} \longleftrightarrow \text{yoy}^+\text{e}$$
$$\text{йи}^*\text{ой}^*\text{е} \longleftrightarrow \text{yi}^+\text{oy}^+\text{e}$$

$$\text{ий}^*\text{ы} \longleftrightarrow \text{iy}^*\text{ea}$$
$$\text{ьх}^*\text{а} \longleftrightarrow \text{ieh}^+\text{a}$$
$$\text{ий}^*\text{ех}^*\text{а} \longleftrightarrow \text{iy}^+\text{eh}^+\text{a}$$

Step 8 (resolution of the vowel collisions: the triple collision)**.**

There is the collision ё-э-ы remaining to be solved. As before, let us build an infinite table of the colliding «overlays», but now it is not two-, but three-dimensional (the first three layers are in figures 6, 7, 8). The horizontal index of a cell is the quantity of the first delimiter, the vertical index – of the second one, the layer index – of the third one.

The form of the layers beginning from the second one repeats the form of a table for a double collision. On the first layer there are no English strings at all, Russian ones fill all the cells except of the first row. This complicated form appear because the rules with the first and the second and the second and the third heads cannot be applied at the same time, but with the first and the third heads – can be (see the first column of the first layer).

			…
ёы	йоы	йиоы	…
ёйы	йойы	йиойы	…
⋮	⋮	⋮	

Figure 6: the table of «overlays», layer 1

	йэа	йиэа	…
ёеа	йоеа yoea	йиоеа yioea	…
ёйеа	йойеа yoyea	йиойеа yioyea	…
⋮	⋮	⋮	

Figure 7: the table of «overlays», layer 2

	йэха	йиэха	…
ёеха	йоеха yoeha	йиоеха yioeha	…
ёйеха	йойеха yoyeha	йиойеха yioyeha	…
⋮	⋮	⋮	

Figure 8: the table of «overlays», layer 3

ёы	йоы	йиоы	…
ёйы	йойы	йиойы	…
ёййы	йоййы	йиоййы	…
⋮	⋮	⋮	

Figure 9: resolution of the collision, layer 1

йэа	йиэа	йииэа	…
ёеа	йоеа yoea	йиоеа	…
ёйеа	йойеа yoyea	йиойеа yioyea	…
⋮	⋮	⋮	

Figure 10: resolution of the collision, layer 2

йэха	йиэха	йииэха	…
ёеха	йоеха yoeha	йиоеха yioeha	…
ёйеха	йойеха yoyeha	йиойеха yioyeha	…
⋮	⋮	⋮	

Figure 11: resolution of the collision, layer 3

In order to resolve the collisions, proceed as follows. We shift the whole first layer by one cell up; in all subsequent layers, we shift the top row by one cell to the left. Now we pair every Russian string with the English string to the bottom right on the next layer from it (figures 9, 10, 11). All the obtained rules are added to Γ.

The natural structure of the set of added rules is three-dimensional. It can be represented in the (*,+)-notation:

$$\text{ёй}^*\text{ы} \longleftrightarrow \text{yoy}^*\text{ea}$$
$$\text{йи}^*\text{ой}^*\text{ы} \longleftrightarrow \text{yi}^+\text{oy}^*\text{ea}$$
$$\text{йи}^*\text{эх}^*\text{а} \longleftrightarrow \text{yi}^*\text{oeh}^+\text{a}$$

$$\text{ёй}^*\text{ех}^*\text{а} \longleftrightarrow \text{yoy}^+\text{eh}^+\text{a}$$
$$\text{йи}^*\text{ой}^*\text{ех}^*\text{а} \longleftrightarrow \text{yi}^+\text{oy}^+\text{eh}^+\text{a}$$

Note that the set of added rules is described by almost the same pattern that the resolution of a double unilateral collision: immediately before a use of a rule with a head symbol, a delimiter is not required. As an elaboration, the word immediately added. Examples:

1) Γ(ёы) = yoea, Γ(ёйы) = yoyea: the rules with the first (ё) and the third (ы) heads are used at the same time, the delimiter й/y is not required before the third one;
2) Γ(йоы) = yioea, Γ(йиоы) = yiioea, Γ(йойы) = yioyea, Γ(йиойы) = yiioyea: the rule with the head ы is used, the second delimiter (й/y, which is immediately before that) is not required, but the first one (и/i) is;
3) Γ(йэа) = yoeha, Γ(йиэа) = yioeha, Γ(йэха) = yoehha, Γ(йиэха) = yioehha: the rule with the head э is used, the first delimiter is not needed, the third one is;
4) Γ(ёеа) = yoyeha, Γ(ёйеа) = yoyyeha, Γ(ёеха) = yoyehha, Γ(ёйеха) = yoyyehha: the rule with the head ё is used, the second and the third delimiters are required;
5) Γ(йоеа) = yioyeha, Γ(йиойеха) = yiioyyehha: no rules with heads, all the delimiters are required.

To sum up, the following law is true.

Law 1. In order to complete the translation in case of a unilateral collision (ё-у, ё-ъ, ё-э, э-ы, ь-ы or ё-э-ы), the usual rules from the colliding chains have to be applied, with the only difference that a delimiter is not required immediately before a use of a rule with a head symbol.

Step 9 (q)**.**

For all the Russian letters, rules containing them in their left parts have been added. But there are three English letters remaining which do not have rules yet: q, w и x.

We introduce the rule къ ⟷ q. As before, we need to add an infinite chain of rules to keep Γ bijective. The issue is symmetric to the known one: there is one more delimiter symbol in the left parts, not in the right ones. Let г/g be the delimiter.

$$\text{къ} \longleftrightarrow \text{q}$$
$$\text{кг}^+\text{ъ} \longleftrightarrow \text{kg}^*\text{oa}$$

Step 10 (x, resolution of collisions)**.**

For the letter x we add the following rules:

$$\text{кс} \longleftrightarrow \text{x}$$
$$\text{кг}^+\text{с} \longleftrightarrow \text{kg}^*\text{s}$$

Note that collisions have emerged between the chains for x and ш, x and щ. For a start consider the collision x-ш (figure 12).

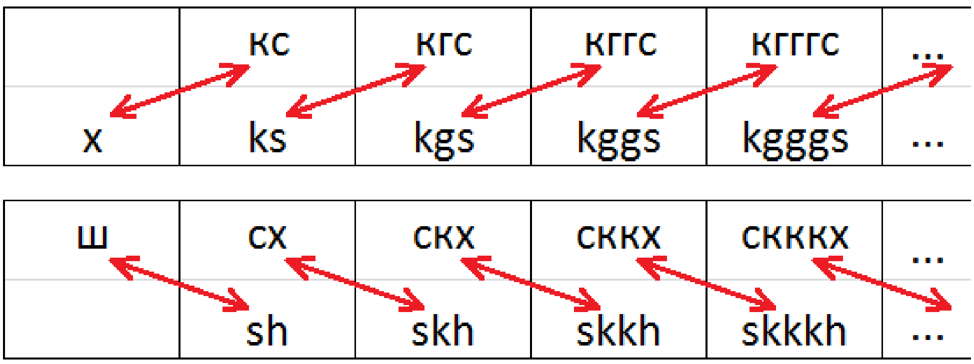

Figure 12: the chains of rules for x and ш

The situation resembles the known one (for example, the collision ё-у), but now one head symbol is English, another is Russian. We call such a collision *bilateral*.

As before, we list all «overlays» into an infinite table (figure 13); the horizontal index of a cell is the quantity of the first delimiter, the vertical index – of the second one. Russian strings fill the whole table except of the first column, English ones – except of the first row. In order to resolve the collision, we pair every Russian string with the English one to the bottom left from it (figure 14). The obtained rules are added to Γ.

	кш	кгш	...
xh	ксх ksh	кгсх kgsh	...
xkh	кскх kskh	кгскх kgskh	...
⋮	⋮	⋮	

Figure 13: the table of «overlays»

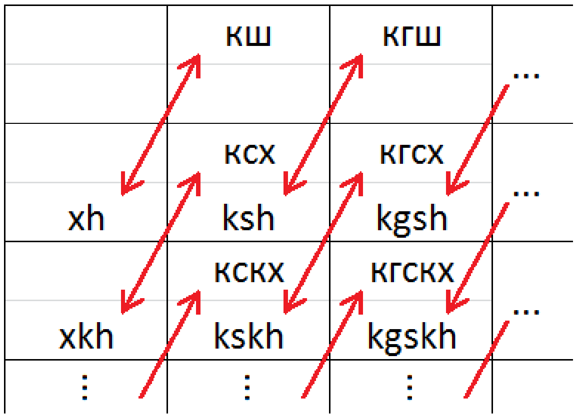

Figure 14: resolution of the collision

The set of added rules can be represented in the (*,+)-notation:

$$\text{кш} \longleftrightarrow \text{xh}$$
$$\text{кг}^{+}\text{ш} \longleftrightarrow \text{kg}^{*}\text{sh}$$
$$\text{кск}^{*}\text{х} \longleftrightarrow \text{xk}^{+}\text{h}$$
$$\text{кг}^{+}\text{ск}^{*}\text{х} \longleftrightarrow \text{kg}^{*}\text{sk}^{+}\text{h}$$

It is easier to comprehend it via the following pattern. Note that for translation of any string from the table, a replacement by one string forward in the chain with a head in the source language is required, then the letter-wise translation and, finally, a replacement by one string backward in the chain with a head in the target language. Examples:

1) Γ(кш) = xh: кш ⟶ (a replacement forward in the chain of ш) ксх ⟶ ksh ⟶ ⟶ (a replacement backward in the chain of x) xh; the reverse translation: xh ⟶ ⟶ (a replacement forward in the chain of x) ksh ⟶ ксх ⟶ (a replacement backward in the chain of ш) кш;
2) Γ(кгш) = ksh: кгш ⟶ кгсх ⟶ kgsh ⟶ ksh; the reverse one: ksh ⟶ kgsh ⟶ ⟶ кгсх ⟶ кгш;
3) Γ(ксх) = xkh: ксх ⟶ кскх ⟶ kskh ⟶ xkh; the reverse one: xkh ⟶ kskh ⟶ ⟶ кскх ⟶ ксх;
4) Γ(кхсх) = kskh: кхсх ⟶ кхскх ⟶ khskh ⟶ kskh; the reverse one: kskh ⟶ khskh ⟶ ⟶ кхскх ⟶ кхсх.

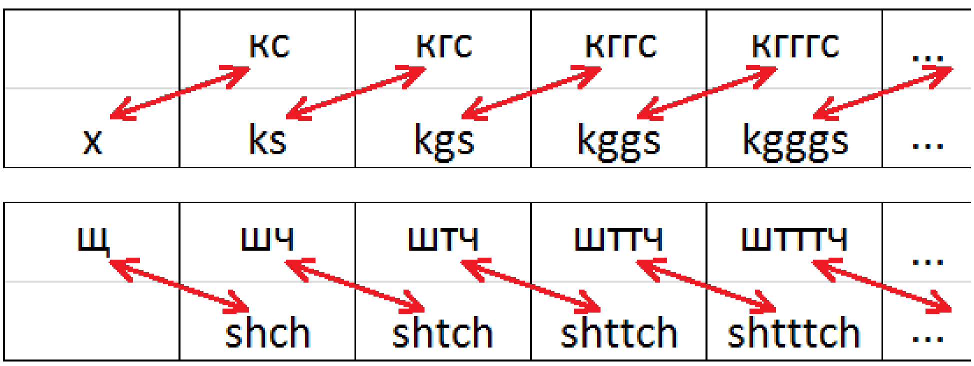

Figure 15: the chains of rules for х and щ

The collision of chains х и щ has a similar form (figure 15). It is resolved likewise, the same pattern is true.

The added rules can be represented in the (*,+)-notation:

$$\text{кщ} \longleftrightarrow \text{xhsh}$$
$$\text{кг}^{+}\text{щ} \longleftrightarrow \text{kg}^{*}\text{shch}$$
$$\text{кшт}^{*}\text{ч} \longleftrightarrow \text{xht}^{+}\text{ch}$$
$$\text{кг}^{+}\text{шт}^{*}\text{ч} \longleftrightarrow \text{kg}^{*}\text{sht}^{+}\text{ch}$$

Note the following feature. When Russian strings from the chains «intersect», «at the junction» there is ш (sh «collapses» into it). That is, for example, кс and шч can «overlay» in кшч, although at first glance it seems false. Also the rules for ш and ч introduced before must be accounted in the letter-wise translation. Examples:

1) Г(кщ) = xhch: кщ ⟶ (a replacement forward in the chain of щ) кшч ⟶ kshch ⟶ ⟶ (a replacement backward in the chain of х) xhch; the reverse translation: xhch ⟶ ⟶ (a replacement forward in the chain of х) kshch ⟶ кшч ⟶ (a replacement backward in the chain of ш) кщ;
2) Г(кгщ) = kshch: кгщ ⟶ кгшч ⟶ kgshch ⟶ kshch; the reverse one: kshch ⟶ ⟶ kgshch ⟶ кгшч ⟶ кгщ;
3) Г(кшч) = xhtch: кшч ⟶ кштч ⟶ kshtch ⟶ xhtch; the reverse one: xhtch ⟶ ⟶ kshtch ⟶ кштч ⟶ кшч;
4) Г(кгшч) = kshtch: кгшч ⟶ кгштч ⟶ kgshtch ⟶ kshtch; the reverse one: kshtch ⟶ ⟶ kgshtch ⟶ кгштч ⟶ кгшч.

To sum up, the following law is true.

Law 2. In order to complete the translation in case of a bilateral collision (х-ш or х-щ), a replacement by one string forward in the chain with a head in the source language is required, then the letter-wise translation and, finally, a replacement by one string backward in the chain with a head in the target language.

Step 11 (w, resolution of a collision)**.** We introduce the following rules for w:

$$\text{ув} \longleftrightarrow \text{w}$$
$$\text{уф}^{+}\text{в} \longleftrightarrow \text{ouf}^{*}\text{v}$$

A collision (figure 16) has emerged between this chain and йи*у ⟷ yi*ou (the rules from the upper row of the table in figure 4).

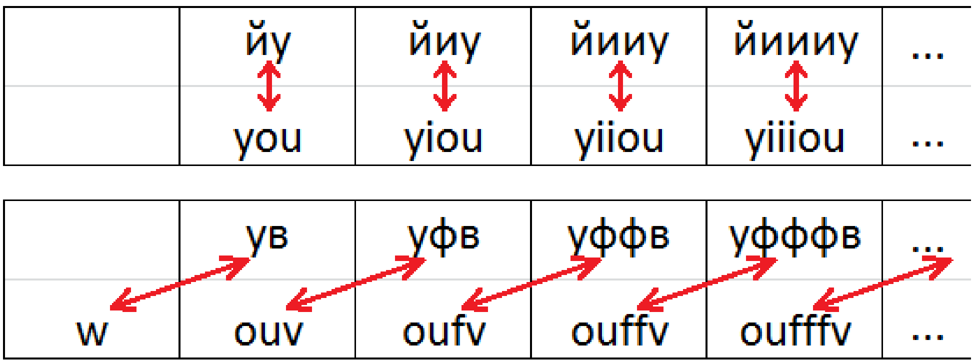

Figure 16: the chains of rules for йу and w

In the chain йи*у ⟷ yi*ou, the strings match letter-wise in some sense; it does not have a regular form of a chain of rules, which is obtained by a shift of the letter-wise matching by one position and adding a head. This collision has only one head (w) – we denote it as йу-w.

Let us build an infinite table of «overlays» (figure 17). Russian strings fill the whole table except of the first row, English strings fill the whole table. In order to resolve the collisions, we shift all the Russian strings by one cell up and pair the strings in one cell with each other.

The added rules can be represented in the (*,+)-notation:

$$\text{йи}^*\text{ув} \longleftrightarrow \text{yi}^*\text{w}$$
$$\text{йи}^*\text{уф}^+\text{в} \longleftrightarrow \text{yi}^*\text{ouf}^*\text{v}$$

 yw	 yiw	 yiiw	...
йув youv	йиув yiouv	йииув yiiouv	...
йуфв youfv	йиуфв yioufv	йииуфв yiioufv	...
⋮	⋮	⋮	

Figure 17: the table of «overlays»

йув yw	йиув yiw	йииув yiiw	...
йуфв youv	йиуфв yiouv	йииуфв yiiouv	...
йуффв youfv	йиуффв yioufv	йииуффв yiioufv	...
⋮	⋮	⋮	

Figure 18: resolution of the collision

As can be seen, a simple pattern is true: the translation is the same as if the rules from the chain of йу are ignored.

Law 3. In case of the collision йу-w, the translation is completed ignoring the chain of йу.

7 Final formulations

The following set Γ of rules from $\mathcal{R}^*$ to $\mathcal{A}^*$ has been built. For convenience let us represent it as $\Gamma = \Gamma_1 \cup \Gamma_2$: in Γ_1 there are the key rules, in Γ_2 – the rules added to resolve collisions.

Γ_1:

а ⟷ a	ж ⟷ j	м ⟷ m	т ⟷ t
б ⟷ b	з ⟷ z	н ⟷ n	ф ⟷ f
в ⟷ v	и ⟷ i	о ⟷ o	х ⟷ h
г ⟷ g	й ⟷ y	п ⟷ p	ц ⟷ c
д ⟷ d	к ⟷ k	р ⟷ r	ю ⟷ u
е ⟷ e	л ⟷ l	с ⟷ s	
тф ⟷ th	тх ⟷ tf	пф ⟷ ph	пх ⟷ pf
ч ⟷ ch	у ⟷ ou	ъ ⟷ oa	къ ⟷ q
цк*х ⟷ ck$^+$h	ой*ю ⟷ oy$^+$u	ох*а ⟷ oh$^+$a	кг$^+$ъ ⟷ kg*oa
ш ⟷ sh	э ⟷ oe	ы ⟷ ea	кс ⟷ x
ск*х ⟷ sk$^+$h	ой*е ⟷ oy$^+$e	ех*а ⟷ eh$^+$a	кг$^+$с ⟷ kg*s
щ ⟷ shch	ё ⟷ yo	ь ⟷ ie	ув ⟷ w
шт*ч ⟷ sht$^+$ch	йи*о ⟷ yi$^+$o	ий*е ⟷ iy$^+$e	уф$^+$в ⟷ ouf*v
	я ⟷ ya		
	йи*а ⟷ yi$^+$a		

Γ_2:

йи*у $\longleftrightarrow$ yi*ou
ёй*ю $\longleftrightarrow$ yoy$^+$u
йи*ой*ю $\longleftrightarrow$ yi$^+$oy$^+$u

ой*ы $\longleftrightarrow$ oy*ea
эх*а $\longleftrightarrow$ oeh$^+$a
ой*ех*а $\longleftrightarrow$ oy$^+$eh$^+$a

йи*ъ $\longleftrightarrow$ yi*oa
ёх*а $\longleftrightarrow$ yoh$^+$a
йи*ох*а $\longleftrightarrow$ yi$^+$oh$^+$a

ий*ы $\longleftrightarrow$ iy*ea
ьх*а $\longleftrightarrow$ ieh$^+$a
ий*ех*а $\longleftrightarrow$ iy$^+$eh$^+$a

йи*э $\longleftrightarrow$ yi*oe
ёй*е $\longleftrightarrow$ yoy$^+$e
йи*ой*е $\longleftrightarrow$ yi$^+$oy$^+$e

ёй*ы $\longleftrightarrow$ yoy*ea
йи*ой*ы $\longleftrightarrow$ yi$^+$oy*ea
йи*эх*а $\longleftrightarrow$ yi*oeh$^+$a
ёй*ех*а $\longleftrightarrow$ yoy$^+$eh$^+$a
йи*ой*ех*а $\longleftrightarrow$ yi$^+$oy$^+$eh$^+$a

кш $\longleftrightarrow$ xh
кг$^+$ш $\longleftrightarrow$ kg*sh
кск*х $\longleftrightarrow$ xk$^+$h
кг$^+$ск*х $\longleftrightarrow$ kg*sk$^+$h

кщ $\longleftrightarrow$ xhsh
кг$^+$щ $\longleftrightarrow$ kg*shch
кшт*ч $\longleftrightarrow$ xht$^+$ch
кг$^+$шт*ч $\longleftrightarrow$ kg*sht$^+$ch

йи*ув $\longleftrightarrow$ yi*w
йи*уф$^+$в $\longleftrightarrow$ yi*ouf*v

Theorem 3. *The set Γ denoted above and also Γ^{-1} are correct systems of rules. Γ defines an English-Russian bijection $F : \mathcal{R}^* \longrightarrow \mathcal{A}^*$, Γ^{-1} defines the opposite function F^{-1}.*

Proof. By theorem 1 both Γ and Γ^{-1} are correct systems of rules. $F(\omega) = \Gamma(\omega)$ is a bijective function from $\mathcal{R}^*$ to $\mathcal{A}^*$ (according to its construction in the previous section). By theorem 2 the system of rules Γ defines the opposite function F^{-1}. □

In order to obtain $\Gamma(\xi)$ by given $\xi \in \mathcal{R}^*$ and $\Gamma^{-1}(\eta)$ by given $\eta \in \mathcal{A}^*$ algorithm 1 can be used. If translating on a computer, a transliteration program can operate the rules of Γ represented in the (*,+)-notation or in some other way.

For transliteration by hand, it is much easier to use the set of key rules Γ_1 and laws 1, 2 and 3, formulated in the previous section. In this case, in algorithm 1 a maximal segment is selected with taking into account the «overlays» of collisions. In the previous section, it is shown that the application of the three laws is equivalent to the use of the rules of Γ_2.

8 Transliteration of a text

Let us shortly consider a task of applied transliteration. Let $\mathcal{R}^c$ be the alphabet of capital Russian letters, $\mathcal{A}^c$ – of capital English letters; $\mathcal{Z}$ is the alphabet of external symbols.

Definition 18. The alphabet $\mathcal{R} \cup \mathcal{R}^c \cup \mathcal{Z}$ is *the extended Russian alphabet*, the language $(\mathcal{R} \cup \cup \mathcal{R}^c \cup \mathcal{Z})^*$ of all the strings on it is *the extended Russian language*. Similarly, $\mathcal{A} \cup \mathcal{A}^c \cup \mathcal{Z}$ is *the extended English alphabet*, $(\mathcal{A} \cup \mathcal{A}^c \cup \mathcal{Z})^*$ is *the extended English language*. We call a string of an extended language *a text*.

It is required to construct a bijection $G : (\mathcal{R} \cup \mathcal{R}^c \cup \mathcal{Z})^* \longrightarrow (\mathcal{A} \cup \mathcal{A}^c \cup \mathcal{Z})^*$ based on the English-Russian bijection $F : \mathcal{R}^* \longrightarrow \mathcal{A}^*$. This can be done simply and naturally as follows.

Definition 19. *A word* on an extended alphabet $\mathcal{R} \cup \mathcal{R}^c \cup \mathcal{Z}$ (or $\mathcal{A} \cup \mathcal{A}^c \cup \mathcal{Z}$) is a finite sequence of symbols from $\mathcal{R} \cup \mathcal{R}^c$ (from $\mathcal{A} \cup \mathcal{A}^c$ respectively) in which all symbols are lower-case, except, perhaps, the first one, which can be upper-case.

Proposition 7. *Any text on $\mathcal{R} \cup \mathcal{R}^c \cup \mathcal{Z}$ (or $\mathcal{A} \cup \mathcal{A}^c \cup \mathcal{Z}$) can be represented as a concatenation of the least possible number of words and sequences of symbols from $\mathcal{Z}$ in exactly one way.*

Algorithm 2 (of transliteration of a text)**.** Let $\tau \in (\mathcal{R} \cup \mathcal{R}^c \cup \mathcal{Z})^*$ be given; we are to obtain $G(\tau)$.

1. Do the decomposition of τ described in proposition 7.
2. Let ω be a word included by the decomposition. Replace the first symbol with a corresponding lower-case one, get the translation $F(\omega)$, then replace the first symbol with an upper-case one if it was upper-case before.
3. Concatenate all the transliterated words and the sequences of symbols on $\mathcal{Z}$ remaining unchanged. The image $G(\tau)$ is obtained.

Completely similarly, $G^{-1}(\nu)$ can be obtained by given $\nu \in (\mathcal{A} \cup \mathcal{A}^c \cup \mathcal{Z})^*$. It is easy to establish that the constructed function G is bijective.