

Introduction to Proto-Indo-European Lexicon Pilot 1.0



Welcome to *Proto-Indo-European Lexicon Pilot 1.0*, the demo version of the generative etymological dictionary of Indo-European languages at <http://pielexicon.hum.helsinki.fi>.

ABSTRACT

Proto-Indo-European Lexicon Pilot 1.0 is the demo version of the Proto-Indo-European Lexicon (PIELex), the generative etymological dictionary of Indo-European languages.

The data of PIE Lexicon Pilot 1.0 is comprised of a PIE root $\sqrt{kahu-}$, $\sqrt{gañu-}$ 'schlagen, usw.'. The data has been chosen in a such manner that it contains examples of all twelve established sub-branches of the Indo-European languages (viz. Albanian, Anatolian, Armenian, Baltic, Celtic, Germanic, Greek, Indo-Aryan, Iranian, Italic, Slavic and Tocharian). In this manner, the data can be understood as a representative small-scale model of the Indo-European languages and the Proto-Indo-European parent language (PIE).

In addition, all classical Indo-European sound laws revised in PYYSALO (2013) appear in the data at least once, which provides a digital proof for the revisions: The entire derivation of Indo-European forms has been digitized by means of *foma*, a programming language developed by MANS HULDEN. Consequently the Indo-European forms of PIE Lexicon are automatically generated from the proto-language.

The success rate of predictions of the PIE Lexicon is currently around 99.9 per cent. Furthermore, the defects (circa 30, marked with red) are generic (i.e. they represent well-defined classes of open research problems of the study, in particular the PIE accent/tone system and minor sound law problems of the individual Indo-European languages).

All in all, this implies that the main bulk of the problems in the reconstruction of the Proto-Indo-European parent language have been effectively solved, except for the PIE accent/tone system and lesser problems already mentioned.

Once a comprehensive system upgrade to PIE Lexicon Pilot 1.1 has been accomplished, the Proto-Indo-European Lexicon will become operational. After that, the project will begin to publish

the PIE Lexicon, the generative etymological dictionary, by uploading large systematic sets of data until the main body of the hundred or so most ancient languages are digitally represented.

The postulation (reconstruction), method and methodologies applied are restricted to the standard practices of natural science. Due to the robust proof procedure of the PIE Lexicon, it is not an exaggeration to assert that a new branch of natural science, comparative Indo-European linguistics, has come to light with the publication of PIE Lexicon Pilot 1.0.

1 A brief history of Indo-European linguistics

1.1 Historical background

1.1.1 After SIR WILLIAM JONES (1788) announced in 1786 the existence of a genetic relationship between Indo-European languages, the pioneers RASMUS RASK and FRANZ BOPP confirmed the existence of systematic correspondences between the “letters” of the Indo-European languages. By the mid-19th century, AUGUST SCHLEICHER was able to conclude that Indo-European linguistics was a branch of natural science as regards its method (viz. comparison). SCHLEICHER’s ideas marked a definite zenith of the study, since he was the first to understand the regularity of sound laws, to invent reconstruction and to sketch out the decision method of Indo-European etymology.

1.1.2 Despite the enormous success, the ideas of RASK, BOPP, SCHLEICHER and their contemporaries – often referred to as the *Paleogrammarians* – contained a cause for rebellion, since they still viewed Sanskrit as the proto-language. This assumption no longer made sense to the following generation of scholars – KARL BRUGMANN, AUGUST LESKIEN, HERMANN OSTHOFF and KARL VERNER – better known as *die Junggrammatiker* (the *Neogrammarians*). This new school squarely concluded that the Sanskrit phoneme inventory did not coincide with that of Proto-Indo-European and, consequently, as BRUGMANN and OSTHOFF stated in their *Manifesto* (1878), a complete revision of the reconstruction was necessary.

1.1.3 Barely after this new doctrine had been established, a Czech scholar BEDŘICH HROZNÝ (1917) proved that Hittite also belonged to the Indo-European language family. The consequences of this became apparent a decade later, when JERZY KURYŁOWICZ, HERMANN MØLLER and EDGAR STURTEVANT independently announced a sensational conclusion: Hittite preserved a (segmental) laryngeal, the phoneme Hitt. *h*, which had been lost in other Indo-European languages – and was hence absent in the Neogrammarian phoneme inventory. This left the PIE reconstruction once more in need of revision.

1.2 The Anatolian laryngeal (Hitt. ḫ) and its competing interpretations

1.2.1 In a series of articles in the 1920s, JERZY KURYŁOWICZ set to work on interpreting the Old Anatolian laryngeal. Unfortunately, his attempt (summarized in KURYŁOWICZ 1935) was not based on the data or its comparison, but on HERMANN MØLLER's Indo-Semitic hypothesis (1906).

Influenced by the biblical idea of a genetic relationship between Indo-European and the Semitic languages, MØLLER suggested the existence of multiple "laryngeals" (*E A O in Proto-Indo-European) and believed that the Proto-Semitic root shape CäCä·(Cä) could be applied to the Indo-European languages as well. These ideas constitute the core of today's mainstream approach, the Laryngeal Theory, in which the notations *h₁ h₂ h₃ are preferred for laryngeals and schwebeablauting root shapes C₁eC₂·C₃ and C₁C₂·eC₃.

1.2.2 In parallel to the emergence of the Laryngeal Theory, the inductive tradition of the Neogrammarians continued. In 1951, LADISLAV ZGUSTA claimed the existence of a single laryngeal in Hittite (and in Old Anatolian), marking the birth of a competitor to MØLLER's Semitic typology. ZGUSTA's theory was subsequently accepted and made known to the Indo-European community by Oswald Szemerényi (1970). Despite the favourable opinion of prominent Anatolian linguists like EMMANUEL LAROCHE (1986) and JOHANN TISCHLER (1977), the theory was criticized by HEINER EIHCNER (1988) for its absence of an own theory. Indeed, it has to be admitted that regardless of the inductive and empirical basis of "monolaryngealism", the early theory remained sketchy: no explanation for the Proto-Indo-European ablaut and the Indo-European vowel patterns was offered. In addition, the Indo-European sound laws were not critically revised and tested, despite the insertion of a new laryngeal phoneme in the PIE phoneme inventory.

1.2.3 These and other defects of the emerging theory were finally remedied by JOUNA PYYSALO (2013), who in his dissertation presented a comparative solution to the laryngeal problem with a single glottal fricative (cover symbol PIE *ḫ = Hitt. ḫ) with a voiceless (PIE *h) and a voiced (PIE *ḥ) variant. In addition to proving the following primary phoneme inventory for Proto-Indo-European

PIE *a/ā? *e/ē *o/ō *i/ī *u/ū *l/ļ *r/ŗ *m/ṃ *n/ņ *k/g *p/b *t/d *h/ḥ *s/z

PYYSALO revised the entire classical sound law system to match the addition of PIE *h/ḥ to the phoneme inventory. In PIE Lexicon Pilot 1.0, it is now verified that this primary phoneme inventory is necessary and sufficient for the derivation of Indo-European forms. Conclusive proof will be sought in future versions by means of complete induction as soon as complete data has been published in the PIE Lexicon.

In addition, the revisions of the Indo-European sound laws presented in PYYSALO 2013 are proven to be correct in PIE Lexicon Pilot 1.0 by means of their digital versions translated into the *foma* programming language (see paragraph 3) in proper chronological order. Each language appearing in the data has a unique foma script of Indo-European sound laws, which automatically generates the Indo-European forms of that language from Proto-Indo-European.

2. A screenshot introduction to PIE Lexicon Pilot 1.0

2.1 Indo-European data, PIE reconstruction, and method in PIE Lexicon Pilot 1.0

2.1.1 The DATA of Proto-Indo-European Lexicon Pilot 1.0 has chosen to satisfy the following conditions in particular:

- (a) All twelve established (uncontested) branches of the Indo-European languages (viz. Albanian, Anatolian, Armenian, Baltic, Celtic, Germanic, Greek, Indo-Aryan, Indo-Iranian, Italic, Slavonic, and Tocharian) are present in the data. In terms of coding, this has enabled us to provide a foma script for all key Indo-European branches.
- (b) All key revisions of PYYSALO 2013 dealing with the PIE phoneme inventory or Indo-European sound law system appear at least once in the data. Consequently, the core of the Indo-European sound law system has been verified in respect to the revisions in Pyysalo 2013.
- (c) The data, comprised of several etymologically connected Indo-European roots, is complete (or nearly so) in terms of attested forms. Accordingly, the data avoids the pitfall of incompleteness.

In this manner, the PIE Lexicon Pilot 1.0 can be understood as a small-scale model of the Proto-Indo-European parent language.

2.1.2 The PROTO-INDO-EUROPEAN PHONEME INVENTORY of PIE Lexicon Pilot 1.0 essentially matches that of Pyysalo 2013 with the replacement of PIE *a for PIE *a in the PIE Lexicon:

*o	*e	*a	*i	*u	*l	*r	*m	*n	*k	*p	*t	*h	*s
* <u>o</u>	* <u>e</u>	* <u>a</u> ?	* <u>i</u>	* <u>u</u>	* <u>l</u>	* <u>r</u>	* <u>m</u>	* <u>n</u>	*g	*b	*d	* <u>h</u>	*z

The phoneme inventory underlines that only these items are allowed in the reconstruction of Proto-Indo-European, and it serves as the table of contents of the future Proto-Indo-European

Lexicon. Once a letter in the PIE Lexicon is published, the respective phoneme of the inventory is changed into an active link allowing navigation within the data.

2.1.3 The PRINCIPLE of POSTULATION, informally FICK'S RULE of "two witnesses", stands as follows:

"Durch zweier Zeugen Mund wird alle Wahrheit kund" — August Fick
(PRINCIPLE of POSTULATION)

In other words, objects at any level can be reconstructed if independently proven by two Indo-European branches (for a detailed explanation, see PYYALO 2013: 1.5.5).

2.1.4 The MAIN ENTRY, the PIE Lexicon root appearing in Pilot 1.0, is shown in the screenshot below.



The main entry contains two root variants and translation. Clicking PIE √kahu- or PIE √gafu- provides a shortcut to the respective entry in the lexicon. The link ([Introduction](#)) refers to this file and ([Abbreviations & References](#)) opens a catalogue of respective items.

2.1.5 Below the main entry, the (P)IE data is managed in two axes, columns and rows, which are discussed separately below.

2.2 The management of data in entry columns

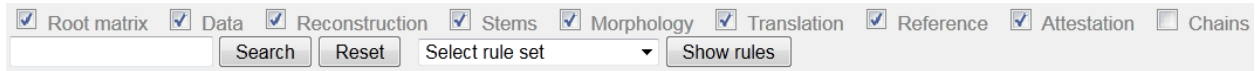
2.2.1 A (DATA) ENTRY always appears in a standard format, consisting of six columns with information, as shown in the screenshot below:

√kahuo-					
PIE *keahyo-	OCS. kovo-	(pr.) 'schmieden, verfertigen'	(Sadnik √374)	(OCS. kovo [1sg], PIE *e for presens)	
PIE *keahyo-	TochB. kawa-	(pref.A.) 'kill'	(DTochB. 208)	(TochB. kawam [1pl], *e for preterite)	
PIE *keahyo-	OSax. hawa-	(pr.) 'hew, cut, strike, smite'	(ASaxD. 534)	(OSax. hawan [inf], *e for presens)	

The first two columns contain PIE roots and their corresponding stems in the Indo-European languages. The third and the fourth columns are reserved for morphological analysis and translations and the fifth for bibliographic references. The actual attested word, its inflectional analysis and optional other information are given in the last column. All data entries can be

hidden/opened by clicking the **Data** button in the CONTROL BAR.

2.2.2 The CONTROL BAR used to manage the data columns and rows is located at the bottom of the website.



Clicking a control button will open/hide the respective data column. The individual data columns can be briefly characterized (from left to right) as follows:

2.2.3 A RECONSTRUCTION (e.g. [PIE *kēāhu-](#)) is the PIE proto-form for the respective Indo-European stem found to its right side. It is also an active link, and clicking it will open a *foma* (SOUND LAW)

CHAIN generating the Indo-European stem. Clicking the control button **Reconstruction** will hide the reconstructions and turn PIE Lexicon into a conventional etymological dictionary without explicit reconstructions for forms.

2.2.4 A (SOUND LAW) CHAIN generated by *foma* can be exemplified by the first data entry yielded by clicking PIE reconstruction:

PIE *kēāhu-	TochA. ko-	(vb.) 'occidere, necare'	(Poucha 85)	(if in TochA. pko [ipv.], HuIdg. 253-)
1. PIE *kēāhu-	PIE *ēa → āa	Colouring rule for *ēa	R2b → *kēāhu-	(Pyysalo 2013: 2.2.10)
2. kēāhu-	PIE *a → Ø	Loss of *a	R7b → *kāhu-	(Pyysalo 2013: 2.2.4)
3. kāhu-	*h → Ø	Loss of segmental *h	R15b → *kāu-	(Pyysalo 2013: 2.1.4)
4. kāu-	*āu → au	Orthographic shortening of āu	R21d → *kau-	
5. kau-	*auC → ōC	Osthoff's Law for auC	R22b → *kō-	(Pyysalo 2013:2.5.8.2)
6. kō-	*ō → o	Orthographic change of ō into o	R27b → TochA. ko-	

The four rows indicate successive sound laws, leading to the Indo-European stem in a manner detailed in Section 3 of this Introduction. Clicking the proto-form again will hide the chain once more. In addition, the CONTROL BAR button **Chains** will open/close all chains simultaneously.

2.2.5 A STEM is a standard entry of the PIE Lexicon, being in practice an Indo-European form without inflectional endings, such as [Lith. káu-](#) (for the abbreviations used for the languages, click [Abbreviations & References](#) on the desktop). The stems are deactivated by clicking the

Stems button in the CONTROL BAR, which turns PIE Lexicon Pilot 1.0 into a Proto-Indo-European dictionary without Indo-European stems on the desktop.

2.2.6 MORPHOLOGY of a stem appears in the column to the right side of the stem. In this way, for example, (vb.) designates a verbal stem (for the morphological abbreviations, click

[Abbreviations & References](#)). This feature can be deactivated with the Morphology button in the CONTROL BAR.

2.2.7 The TRANSLATION (or semantics) of a stem (e.g. ‘[schlagen, hauen, umbringen, vernichten](#)’) is usually provided in the language used by the primary reference. The translations can be hidden/opened from the Translation button in the CONTROL BAR.

2.2.8 A REFERENCE, for example ([LiEtWb. 232](#)), identifies the scientific source from which the attested form has been quoted. The references can be hidden/opened from the Reference button in the CONTROL BAR.

2.2.9 An ATTESTATION, for example ([Lith. kãuti \[inf.\]](#)), denotes an inflected Indo-European form optionally including grammatical analysis and other relevant information. This feature can be hidden/opened from the Attestation button in the CONTROL BAR.

2.3 Management of data in the rows of the root matrix

2.3.1 A ROOT MATRIX corresponds to the column of Proto-Indo-European roots (symbol: $\sqrt{\quad}$) subordinated to the MAIN ENTRY (here PIE $\sqrt{\text{kahu-}}$, $\sqrt{\text{gafu-}}$). The root matrix can be hidden/opened by clicking the Root matrix button in the CONTROL BAR. In addition, the structure of the root matrix as such becomes visible by clicking the Data button in the CONTROL BAR.

2.3.2 A NODE of a ROOT MATRIX refers to a PIE single root of the matrix, comprised of subordinated Indo-European stems morphologically ordered with regard to the extension (or its absence), as indicated in the screenshot below:

$\sqrt{\text{kahu-}}$			(IEW. 535 *kəu-)	(WP. 1:330f.)
PIE *kēchu-	Lith. kãu-	(vb.) ‘schlagen, hauen, umbringen, vernichten’	(LiEtWb. 232)	(Lith. kãuti [inf.])
PIE *kēchu-	Lith. kãu-	(vb.refl.) ‘sich schlagen, kämpfen, ausgelassen sein, tollern’	(LiEtWb. 232)	(Lith. kãutis [inf.])
PIE *kēchu-	Latv. kaũ-	(vb.) ‘schlagen, hauen, stechen, schlachten, treten’	(LiEtWb. 232)	(Latv. kaũt, kavu [pret], kãvu [pret])
PIE *TI-kēchu-	Lith. nu-kãu-	(vb.) ‘erschlagen : kill’	(Senn 2:227)	(Lith. nu-kãuti [inf.])
PIE *TI-kēchu-	Latv. nũo-kaũ-	(vb.) ‘erschlagen, tötēn’	(IEW. 535)	(Latv. nũokaũt [inf.])
PIE *kēchu-	Pol. ku-	(pr.) ‘schmieden’	(LiEtWb. 232)	(Pol. kuć [inf.], kuję [1sg])
PIE *kēchu-	TochA. kau-	(vb.) ‘= Skt. vadhãya- : tötēn’	(DTochB. 208)	(TochB. kausi-s [inf.])
PIE *kēchu-	TochA. kãw-	(vb.) ‘occidere, necare’	(Poucha 85)	(TochA. kãw-e(ñic) [3pl])
PIE *kēchu-	TochA. ko-	(vb.) ‘occidere, necare’	(Poucha 85)	(if in TochA. pko [ipv.], Huldg. 253-)

2.3.3 A ROOT NODE refers to a PIE root (here $\sqrt{\text{kahu-}}$) governing the node and (optional) etymological sources related to it, as seen with “(IEW 535 *kəu-)” in the screenshot below:

A DATA NODE refers to the material subordinated to a single ROOT NODE. Both are referred to by the term 'NODE'.

2.3.4 A CONJECTURE, usually an etymology not available in the general literature, is indicated by adding relevant information to the upper-right corner of a ROOT NODE. For example:

	√gafuvers-			(Pyysalo)
PIE *gafuvers-	Hitt. gūrš-	(vb1.) '(ab)schneiden'	(HHand. 81)	(Hitt. ku-e-er-šu-un [1sg])
PIE *gafuvers-	OIr. geirr-	(sb.) 'said of grain'	(DIL. 358)	(OIr. geirr)

This shorthand states that Pyysalo takes responsibility for the conjecture first published in relation to this connection. A more detailed account of the new etymological conjectures presented in PIE Lexicon Pilot 1.0 will be published in a PIE Lexicon-affiliated scientific journal as soon as the paperwork and other arrangements required for founding a journal have been completed.

2.4 General features of PIE Lexicon site

2.4.1 The SEARCH window is available in the CONTROL BAR.

As an example, writing 'Lith.' for Lithuanian and clicking the search button will provide the Lithuanian data of Pilot 1.0 (with the three first lines reproduced below):

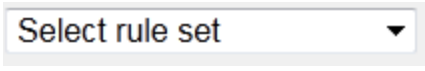
PIE √kahu-, √gafu-	'schlagen, hauen, töten...'	(Introduction)	(Abbreviations & References)
PIE *kəhu-	Lith. káu-	(vb.) 'schlagen, hauen, umbringen, vernichten'	(LiEtWb. 232) (Lith. káuti [inf.])
PIE *kəhu-	Lith. káu-	(vb.refl.) 'sich schlagen, kämpfen, ausgelassen sein, tollern'	(LiEtWb. 232) (Lith. káutis [inf.])
PIE *H-kəhu-	Lith. nu-káu-	(vb.) 'erschlagen : kill'	(Senn 2:227) (Lith. nu-káuti [inf.])
...			

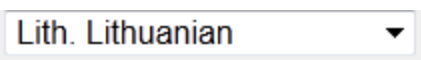
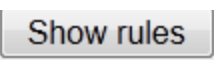
This feature allows use of the PIE Lexicon as a dictionary of individual Indo-European languages.

2.4.2 The KEYBOARD

·	à	á	â	ã	ä	è	é	í	î	ï	ð	ñ	ò	ó	ø	ù	û	ý	þ	ā	ą	č	ē	ě	ę	ī	ĭ	ĭ	ŋ	ō	ś	š	ŭ	ū	ž	ə	q	j	dz		
ō	ε	ζ	η	θ	ι	λ	μ	ν	ο	π	ρ	ς	σ	τ	υ	φ	ψ	ω	ĩ	đ	ƒ	đ	é	ħ	ṃ	ṇ	ṅ	ó	ş	ţ	ě	á	ã	ö	á	é	ı	ó	ú	ä	í

automatically appears the when the cursor is held over the SEARCH window. From there, the complex symbols of PIE and IE languages can be directly obtained.

2.4.3 The SELECT RULE SET window  contains all of the foma scripts of Indo-European languages that are available in Pilot 1.0. For example, selecting

  and clicking SHOW RULES will open the Lithuanian foma script in a new pop-up window in the browser.

2.4.4 A SEGMENT marked in RED indicates an inconsistency between an attested Indo-European form on the desktop and the respective outcome of a foma chain. For instance, a perfectly correct (attested) form of Tocharian appears on the desktop:

PIE *[kēahusjont-](#) ToChA. [košant-](#) (sb.m.) ‘carnifex : hangman, executer’ (Poucha 85 & 88)

Despite the form being correct, the vowel ToChA. /ä/ is marked with red, because the foma rules of Pilot 1.0 yield a vowel ToChA. /a/ instead:

8. [kōšant-](#) *ō → o Orthographic change of ō into o R27b → [ToChA. košant-](#)

The red in the attested form is only added to draw attention to the fact that an open research problem exists around the genesis of the vowel /ä/ in Tocharian. It should not be taken as casting doubt on the correctness of the attested form itself. The PIE Lexicon Project will discuss such discrepancies in the forthcoming journal article on the website (mentioned above), seeking answers to this and similar problems in order to better facilitate automatic generation of all the data.

2.4.5 The LICENCE, found at the bottom of the website, links to where additional information concerning the licence can be found:



The PIE Lexicon Project licenses its data, linguistic results and finite-state encoded rules under a [Creative Commons Attribution-ShareAlike \(CC BY-SA\) license](#).

2.4.6 Finally, it should be noted that PIE Lexicon Pilot 1.0 is the very first version of the Proto-Indo-European Lexicon. Accordingly, all of its properties and features will be considerably improved and supplemented in the future. In addition, multiple defects (such as the ones outlined below) will be eliminated in future versions:

(a) The foma scripts for the sound laws of individual Indo-European languages are not complete, but rather designed to handle the data of Pilot 1.0 (with some extra rules added in order to provide basic sound laws for the languages handled).

(b) In some instances, an ambiguous vowel (e.g. PIE *e or *o) is interpreted as PIE *e or PIE *o instead of a proper cover symbol for an ambiguous (or unproven) proto-phoneme. Similar shortcuts will not be found in the following, more advanced version.

(c) The prefixes Π and suffixes Σ have not been handled. This is not due to any difficulty in generating the forms, but by the absence of complete lexical data in Pilot 1.0 regarding the affixes: in order to demonstrate this, the item PIE * $\text{h}_2\text{epo-gah}_2\text{ueah-}$ \rightarrow Gr. $\text{\u03ac}\pi\text{o}\cdot\text{\u03c6}\alpha\text{-}$ has been reconstructed with a prefix by means of the usual foma rules that apply to roots. Accordingly, as soon as affixes are treated in the PIE Lexicon, these will also be generated.

3 Foma and its extensions in the PIE Lexicon

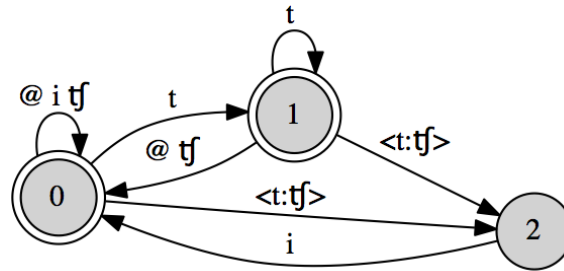
3.0 General introduction to finite-state technology in linguistics

3.0.1 Foma (HULDEN, 2009a) is a compiler for producing finite-state machines out of linguistic descriptions. It is particularly suited for modeling alternation and derivation processes in phonology and morphology.

3.0.2 Finite-state transducers have long been recognized in synchronic linguistics as being suitable computational models for treating sequential derivations of phonological change. The fact that individual phonological rules of the type introduced in the influential *Sound Pattern of English* (CHOMSKY and HALLE 1968) could be modeled as finite-state transducers was first noted by C. DOUGLAS JOHNSON (1972). This discovery went largely unnoticed at the time, but the same observation was made in a brief note by KAPLAN and KAY (1981), which was later elaborated on and refined (KAPLAN and KAY 1994) to produce a complete, formal and computational model for manipulating and efficiently reasoning about the effects of a word undergoing sequences of phonological sound-change rules. The possibilities of using finite-state calculus to model sound change through the combination of finite transducers has since been greatly expanded (KARTTUNEN 1995, KEMPE and KARTTUNEN 1996, YLI-JYRÄ 2008, HULDEN 2009b).

3.0.3 In essence, a finite-state transducer is an abstract computational device for modeling certain classes of relations (called *regular relations*) between sequences of symbols. A transducer is often informally depicted as a directed graph consisting of states joined together by labeled transitions. The set of relations modeled by the transducer is captured by the paths of the graph, when traversing from an initial state (numbered 0) to any terminal state (marked with a double circle). A transducer is an abstract translation device that reads input strings, matches these against the input symbols (to the left of the colon) on the transitions, and exports the corresponding output strings (to the right of the colon). Simple symbol repetitions are also possible (here depicted as the single symbols "t", "i", etc.). In the example transducer below, we

have also used an abstract meta-symbol "@" in the labels to represent repetition of symbols not mentioned elsewhere.



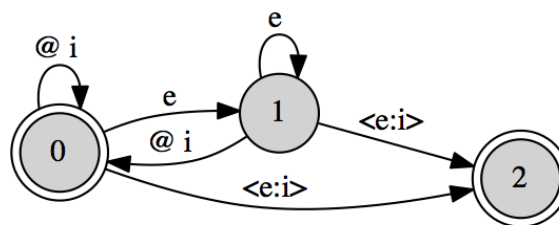
The transducer above accepts among other strings as input /tati/ and translates it into the string [taʧi], following the state path 0-1-0-2-0. Upon further inspection, it can be seen that instead of just being able to translate this one word, the transducer in fact models the very common phonological process of palatalization of [t] before the front high vowel [i]. This can be expressed in a more phonological notation as:

$$t \rightarrow \text{tʃ} / _ i$$

One of the reasons for modeling synchronic phonology with transducers is the possibility of joining several transducers by means of composition. That is, given two or more transducers, one can calculate a new single transducer whose effect on a sequence of symbols is logically equivalent to having passed that sequence consecutively through each transducer. For the sake of illustration, consider another phonological rule, that of word-final vowel raising:

$$e \rightarrow i / _ \#$$

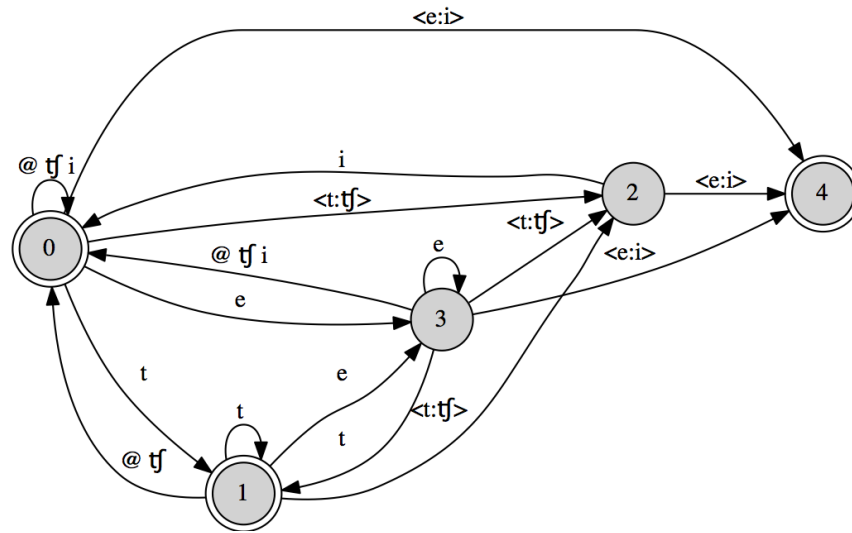
This can be modeled by the transducer:



The effect of applying the above two rules of raising and palatalization in a feeding order where raising applies first, followed by palatalization, is modeled in sequences such as:

- (a) tate
- ↓ (word-final raising)
- (b) tati
- ↓ (palatalization of t)
- (c) taʧi

The two transducers can be composed together to yield a monolithic transducer that captures the rule interaction in one single step.



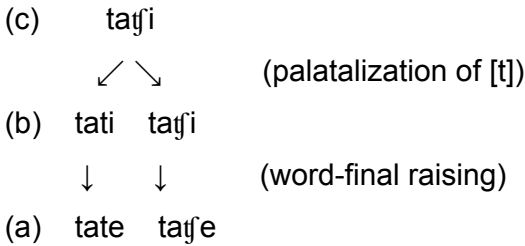
Transducers are inherently bidirectional devices. Given a transducer, it is possible to not only calculate the effect it has on an input form, but also in a trivial manner the inverse relation. Informally, this means being able to automatically determine what possible input forms could have produced a given output form in a system consisting of several consecutive sound changes. For the monolithic transducer for raising-palatalization, feeding the form "taʃi" as the inverse, one can immediately calculate that in this system there are four possible input forms that could have produced the output "taʃi", as follows:

- (c) taʃi
 ↙ ↘ (palatalization of [t])
- (b) tati taʃi
 ↙ ↘ ↙ ↘ (word-final raising)
- (a) tate tati taʃe taʃi

While the conclusion that there are four logical possibilities for an input form corresponding to the output [taʃi] is easily drawn after some reasoning, such reasoning is far from trivial or obvious when dealing with dozens of temporally ordered sound laws. It is here that the transducer model shows its usefulness; we can (1) automatically convert phonological rules to transducers, and (2) automatically calculate all the logical possibilities for any input-output relation defined under this formalism.

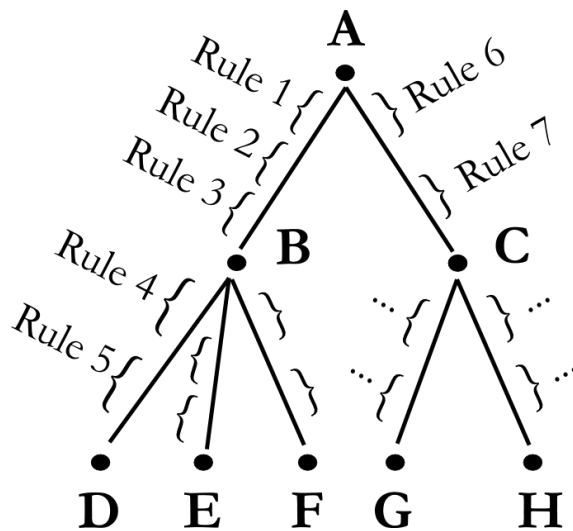
Above we have not constrained the set of inputs to the transducers in any way; they accept arbitrary symbol sequences as input strings and translate them into corresponding outputs as

dictated by their structure. However, finite-state calculus allows us to impose structural constraints on different levels of representation. That is, it is possible to postulate word-structure constraints that describe what the possible words are at each level. In the above example, we could declare that forms (a) never contain a word-final high vowel [i] and, consequently, that all word-final occurrences of [i] at point (b) are a result of raising. This would produce a different system where an inverse mapping of (c) "taʃi" could only correspond to either "taʃe" or "tate" at (a).



3.1 Diachronic modeling with transducers

3.1.0 The same techniques that have been used to model derivational phonological processes in synchronic studies can be put to use in diachronic linguistics. In the diachronic case, we assume that the sound changes in question model sound laws, possibly conditioned by some phonological or morphological environment. We also assume they are given a set of attested forms and sound laws pertinent to the languages in question. From this information, the individual rules can be modeled as transducers, and one can calculate all of the consequences of a single rule or attested form, as well as identify any contradictions or inconsistencies in the formulation of sound laws.



The above figure shows a hypothetical fragment of a language tree, together with some sound laws that are assumed to be temporally ordered. If a transducer model is constructed in a manner consistent with the tree, a number of important questions can be answered immediately and automatically by means of calculus. For example, if we are given an attested form in any one of the languages (e.g. F), we can immediately calculate what the corresponding cognates could logically be in the other languages (i.e. A,B,C,D,E,G,H). This is true even if no proto-form is known or postulated (e.g. language A), although a postulated specific proto-form may serve to constrain the logical possibilities.

3.1.2 For the present purposes (of PIE), the most natural calculation is, of course, to assume a proto-form (A) and from there to automatically derive the cognates in the daughter languages (B,C,D,E,G,H) using the rules; this is done in order to confirm the consistency of the postulated sound laws and proto-forms with attested forms in the languages.

3.2 Summary of finite-state modeling

3.2.1 Applying finite-state transducers to calculate postulated historical sound changes provides a rigorous system of formal calculus for mapping cognates from hypothetical proto-forms to daughter languages and automatically evaluating the consistency of such mappings. In its simplest application, this involves calculating the effect of a sequence of sound laws on a reconstructed form, immediately providing us with an exact prediction of what the cognates in the daughter languages are. However, the calculus itself allows modeling of far more complex relationships, which include:

- Calculating the consistency of attested forms in daughter languages according to a set of postulated sound laws without necessarily ever reconstructing a specific proto-form.
- Employing word-structure constraints at specific points in a phylogenetic tree or network to rule out ambiguous derivations.
- Including effects of (re-)borrowing and analogy at specific nodes in a phylogenetic tree or network.

3.2.2 All of the above are interesting targets of research. In the present pilot project, we have largely constrained ourselves to using transducer technology to calculate cognates from proto-forms.

3.3 The rule format of foma

3.3.1 In the current project, we have used a modification of *foma* finite-state compiler software, which allows us to not only compile phonological rules (sound laws) into finite-state transducers using the formalism provided by *foma*, but also to provide commentary on each individual rule in line with the conventions of Indo-European scholarship, as well as detailed analyses of where and how each rule should apply.

Foma itself is a multipurpose finite state machine compiler that can convert formal language descriptions expressed in various formalisms to finite automata and subsequently manipulate these automata in different ways. Descriptions can be provided in a mathematical formalism (akin to the one employed by KAPLAN and KAY (1994), a logical formalism (HULDEN 2009), or regular expressions of the format presented in BEESLEY and KARTTUNEN (2003) for the Xerox *xfst* tool. In the following, we will largely make use the *xfst*-compatible formalism, with some modifications which are discussed below in 3.4.

3.3.2 The basic phonological rule in *foma* assumes the format:

$$A \rightarrow B \parallel C _ D$$

where A,B,C, and D are individual *regular languages*. In modeling the sound laws, we have limited ourselves to special cases in which each argument is a simple sequence of symbols. The semantics of such rules corresponds roughly to the notion of *obligatory replacement*. That is, any occurrence of a sequence of a symbol sequence A must be replaced by the sequence B if the sequence A occurs in the environment C _ D (i.e. between C and D). The arguments C and D may be empty. Several rules may also be declared to act in parallel using the ,, operator, as in:

$$A \rightarrow B \parallel C _ D \text{ ,, } E \rightarrow F \parallel G _ H$$

Here the two rules are unordered with respect to each other, and they apply simultaneously to an input sequence of symbols. Several ordered rules can be joined together by means of *composition* (.o.), whereby is calculated a transducer that is the equivalent of two rules being applied sequentially; for example:

$$A \rightarrow B \parallel C _ D \text{ .o. } E \rightarrow F \parallel G _ H$$

In general, we try to avoid combining rules with composition, since doing so complicates the task of later recovering in detail how each rule contributes to changing a proto-form into a cognate in the daughter languages.

3.4 Rule comments and order

3.4.0 In the *foma* modification used in PIE Lexicon, we have followed a specific format of formal rule declaration, commentary and rule order. In general, the format of the rules found in the PIE language-specific files is as follows:

```
define RULENUMBER FOMA-RULE ; # SCHOLAR-RULE COMMENTARY
```

Here, the `RULENUMBER` is an arbitrary number for the rule in question, `FOMA-RULE` is the actual sound law (or fragment of it) defined in the format discussed above, `SCHOLAR-RULE` is the same information provided in a more scholarly style, and `COMMENTARY` is a text designed to aid the user of the PIE Lexicon in analyzing the specific order and manner in which rules have been applied to some form. For example, our rule sets contain the following line:

```
define R1a e -> a || a _ ; # PIE *ae → aa Colouring rule for *ae
```

3.4.1 `RULE ORDER`. At the bottom of each language's rule-file, we also find a declaration of the specific order in which rules are to be applied for that language. This is a simple declaration of the form:

```
chain R1, R2, ..., Rn
```

This indicates the specific temporal (or chronological) ordering in which the sound laws should be realized.

4. The Web service of PIE Lexicon (Pilot 1.0.)

4.1 A Web interface can cause slower computers some problems when making display adjustments. The Web service currently show the whole data set on a single page, so it requires available memory and significant computing power to adjust the display settings. The interface will be optimized in the next version; faster and more stable usage can be expected then.

4.2 The backend of the Web service is implemented with [Node.js \(Express\)](#). All of the data models are in [JSON](#) format, supporting easy development, scaling and integration.

4.3 The interface itself is made with a combination of standard HTML, CSS, JavaScript and [jQuery](#).

4.4 If interested in the code behind the Web service, please contact us at pie-lexicon@helsinki.fi.

5 Conclusion/Summary

With the generative capabilities of the PIE Lexicon, Indo-European linguistics is entering into the new millennium with a synthesis of old and new. On one hand, the method is the same as it always was; on the other hand, its digitization brings the study to a form best characterized as a new branch of natural science.

In line with our adventurous, curious and scientific spirit, we welcome all criticism, comments and suggestions that will help us to evaluate and develop the PIE Lexicon for the future. Accordingly, for linguistic, computational or interface issues, please contact the PIE Lexicon Project at pie-lexicon@helsinki.fi.

Proto-Indo-European Lexicon Pilot 1.0 was commissioned by Prof. Arto Mustajoki, Dean of the Faculty of Arts at the University of Helsinki, and completed with the funding of the faculty. Our team wishes to express its great gratitude for assistance in making this critical phase of the project possible.

Jouna Pyysalo (PI), Mans Hulden, Mika Järvinen, and Aleksis Sahala

6. References

- BEESLEY, K. B., and KARTTUNEN, L. (2003). *Finite-state morphology: Xerox tools and techniques*. CSLI, Stanford.
- BRUGMANN, KARL and OSTHOFF, HERMANN (1878). Vorwort. *Morphologische Untersuchungen auf dem Gebiete der indogermanische Sprachen*, Vol. 1. Leipzig: Hirzel, iii–xx.
- CHOMSKY, N., and HALLE, M. (1968). *The Sound Pattern of English*. New York: Harper and Row.
- EICHNER, HEINER (1988). “Anatolisch und Trilaryngalismus”. In: *Die Laryngaltheorie und die Rekonstruktion des indogermanischen Laut- und Formensystems*. Heidelberg: Winter, pp. 123–151.
- HROZNÝ, BEDŘICH (1917). *Die Sprache der Hethiter, ihr Bau und ihre Zugehörigkeit zum indogermanischen Sprachstamm*. (Boghazköi-Studien 1–2). Leipzig: Hinrichs.
- HULDEN, M. (2009a). “Foma: a finite-state compiler and library”. In: *Proceedings of the EACL 2009 Demonstrations Session* (Athens, Greece). Association for Computational Linguistics, pp. 29–32.
- HULDEN, M. (2009b). *Finite-State Machine Construction Methods and Algorithms for Phonology and Morphology*. PhD Dissertation, University of Arizona.
- HULDEN, M. (2009c). Regular expressions and predicate logic in finite-state language processing. In *Proceedings of the 2009 conference on Finite-State Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP*, pp. 82–97.
- JOHNSON, C. D. (1972). *Formal Aspects of Phonological Description*. The Hague: Mouton.
- JONES, SIR WILLIAM (1788). Anniversary Discourse (February 2nd 1786). *Asiatick Researches* 1:415–431.
- KAPLAN, R. and KAY, M. (1981). “Phonological rules and finite-state transducers”. Paper presented to the Winter Meeting of the Linguistic Society of America, New York.
- KARTTUNEN, L. (1995). “The replace operator”. In: *Proceedings of the 33rd annual meeting of the Association for Computational Linguistics* (Cambridge, MA). Association for Computational Linguistics, pp. 16–23.
- KEMPE, A., and KARTTUNEN, L. (1996). “Parallel replacement in finite state calculus”. In: *Proceedings of the 16th conference on Computational linguistics* (Copenhagen, Denmark), Volume 2. Association for Computational Linguistics, pp. 622–627.

- KURYŁOWICZ, JERZY (1935). *Études indoeuropéennes I*. (Polska Akademia Umiejętności. Prace komisji językowej n° 21). Cracovie: Gebethner et Wolff.
- LAROCHE, EMMANUEL (1986). *Les laryngales de l'anatolien: état des questions. Comptes rendus de séances de l'Académie des Inscriptions et Belles-Lettres*. 130e année, N. 1: 134–140.
- MØLLER, HERMANN (1906). *Semitisch und Indogermanisch I (Konsonanten)*. Köpenhavn: Hagerup.
- PYYSALO, JOUNA (2013). *System PIE: The Primary Phoneme Inventory and Sound Law System for Proto-Indo-European*. Academic Dissertation. Publications of the Institute for Asian and African Studies 15. Helsinki, Unigrafia Oy.
- SZEMERÉNYI, OSWALD (1970). *Einführung in die vergleichende Sprachwissenschaft*. Darmstadt: Wissenschaftliche Buchgesellschaft.
- TISCHLER, JOHANN (1977). *Hethitisches etymologisches Glossar. Mit Beiträgen von Günter Neumann und Erich Neu*. (IBS 20.) Innsbruck: Institut für Sprachwissenschaft der Universität Innsbruck.
- YLI-JYRÄ, A. (2008). "Transducers from parallel replace rules and modes with generalized lenient composition". In: *6th International Workshop on Finite-State Methods and Natural Language Processing, FSMNLP 2007* (Helsinki). Revised Papers, pp. 197–212.
- ZGUSTA, LADISLAV (1951). *La théorie laryngale*. *Archiv Oriental* 19: 428–472.