

Computersimulierter Lautwandel:

Vom Urindogermanischen in die altgriechischen Dialekte

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Arts (B.A.)
im Fach Historische Linguistik

Humboldt-Universität zu Berlin
Philosophische Fakultät II
Institut für deutsche Sprache und Linguistik

eingereicht von Angelo Papenhoff
geb. am 24.7.1992
in Berlin

1. Gutachter: Herr Prof. Dr. Wolfgang Hock
2. Gutachterin: Frau Prof. Dr. Anke Lüdeling

Berlin, den 10.8.2015

Inhaltsverzeichnis

1. Abkürzungen und Konventionen	2
2. Einleitung	4
3. Zum Lautwandel allgemein	4
4. Iga	6
4.1. Vorüberlegungen	6
4.2. Reguläre Ausdrücke	6
4.3. Funktionsweise	7
5. Modellierung der Lautentwicklung des Altgriechischen	12
5.1. Vorbemerkungen	12
5.2. Das Lautsystem des Urindogermanischen	13
5.3. Urindogermanisch bis Urgriechisch	14
5.4. Urgriechisch bis mykenische Zeit	26
5.5. Nachmykenische/alphabetische Zeit	28
6. Schluss und Ausblick	39
Literatur	41
A. Quelltext	42
A.1. main.scn	42
A.2. subfuncs.scn	46
A.3. rules.scn	49
A.4. griechisch.scn	52
A.5. run-gr.scn	60

1. Abkürzungen und Konventionen

Reguläre Ausdrücke werden in Schreibmaschinenschrift und zwischen Schrägstriche gesetzt (/regex/). Strings werden in Schreibmaschinenschrift und Anführungszeichen gesetzt ("string"). Scheme-Code wird in Schreibmaschinenschrift gesetzt. Bedeutungen werden in einfache Anführungszeichen gesetzt ('foo') Phoneme werden kursiv zwischen Schrägstriche gesetzt (/a/). Phone bzw. Phonemkandidaten werden kursiv gesetzt; wird die Phonetik explizit betont, stehen Phone in eckigen Klammern ([a]). Grapheme werden in spitzen Klammern gesetzt (<a>). Rekonstrukte werden mit Sternchen markiert (*a). In den formulaischen Lautgesetzen werden rekonstruierte und belegte Laute nicht unterschieden. Die Lautschrift ist an das IPA angelehnt, folgt ihm aber nicht völlig (teilweise um eine Festlegung der exakten Lautung zu vermeiden). Lautgesetzliche Entwicklung wird mit > bzw. < bezeichnet.

Sprachen

ach.	achaisch	kypr.	kyprisch
ahd.	althochdeutsch	lak.	lakonisch
ai.	altindisch	lat.	lateinisch
aiol.	aiolisch	lesb.	lesbisch
aitol.	aitolisch	lit.	litauisch
ark.	arkadisch	lokr.	lokrisch
att.	attisch	meg.	megarisch
av.	avestisch	myk.	mykenisch
boiot.	boiotisch	nwdor.	nordwestdorisch
dor.	dorisch	nwgr.	nordwestgriechisch
el.	elisch	oarg.	ostargolisch
euboi.	euboiisch	ostion.	ostionisch
got.	gotisch	othess.	ostthessalisch
gr.	griechisch	pamph.	pamphylich
hom.	homerisch	phok.	phokisch
idg.	indogermanisch	sardor.	saronisches dorisch
inseldor.	inseldorisch	sdor.	süddorisch
inselion.	inselionisch	sogr.	südostgriechisch
ion.	ionisch	thess.	thessalisch
kor.	korinthisch	warg.	westargolisch
kret.	kretisch	wthess.	westthessalisch

Grammatisch Abkürzungen

A.	Akkusativ	Imp.	Imperativ
m.	maskulinum	L.	Lokativ
Aor.	Aorist	n.	neutrum
D.	Dativ	N.	Nominativ
Du.	Dual	Pass.	Passiv
f.	femininum	Pl.	Plural
G.	Genitiv	Sg.	Singular
I.	Instrumental	St.	Stamm

Coversymbole

C	Konsonant	N	Nasal
D	Media	R	Resonant
H	Laryngal	T	Tenuis
K	Velar	T ^h	Tenuis aspirata
K ^w	Labiovelar	V	Vokal
L	Liquida	Y	Halbvokal

2. Einleitung

Ziel dieser Arbeit ist es, den Lautwandel vom Urindogermanischen in die altgriechischen Dialekte am Computer zu simulieren. Seit den Junggrammatikern im 19. Jh. spricht man in der historisch-vergleichenden Sprachwissenschaft von "Lautgesetzen". Dieser Begriff entsprang dem Bestreben, die Linguistik von der Geisteswissenschaft zur Naturwissenschaft zu machen und suggeriert, dass der Lautwandel einen ähnlichen Status habe wie die Naturgesetze in der Physik: wann immer ein Gesetz wirken kann, wirkt es ohne Ausnahme. Diese Lautgesetze müssen für eine der historischen Entwicklung möglichst nahekommende Simulation präzise und in korrekter Chronologie formuliert werden, was allein schon ein großer Nutzen wäre, da alle Beschreibungen der Lautwandel in der Literatur immer nach Lauten, nie zeitlich sortiert sind. Dies wird gegenüber der traditionellen Arbeit noch dadurch erleichtert, dass sich die Richtigkeit der Lautwandelmodellierung an sicheren Beispielwörtern ohne großen Aufwand überprüfen lässt und umgekehrt sich bei einer gegebenen Modellierung ohne Aufwand die Lautgesetzlichkeit einzelner Wörter feststellen lässt. Beides wäre bei manueller Arbeit fehleranfällig und mühsam; eine Automatisierung ist also äußerst lohnend.

Das hier vorgestellte Programm `lga` ist der Nachfolger eines Prototyps, der ein Skript für das UNIX-Programm `sed` war. Dieser war sehr primitiv und wies viele Mängel auf, zeigte aber, dass sich die Arbeit an einem besseren Programm lohnen würde. Ebenso ist allerdings `lga` nur als Prototyp für ein elaborierteres Programm zu verstehen. Im Laufe der Arbeit wird sich herausstellen, welche Probleme, Mängel und weitere Anforderungen es gibt, die in einer nächsten Version angegangen werden müsste.

Nach einigen methodisch-theoretischen Überlegungen zum Lautwandel werde ich im ersten Teil dieser Arbeit `lga` in seinen Designideen und in seiner Funktionsweise vorstellen. Im zweiten Teil stelle ich eine vorläufige Formulierung und relative (teilweise absolute) Chronologie der Lautgesetze dar, die die Entwicklung vom Urindogermanischen in die altgriechischen Dialekte beschreiben. Hierfür wäre `lga` natürlich prinzipiell nicht nötig gewesen und wird auch im Text nicht weiter erwähnt. Ohne dieses nützliche Werkzeug hätte jedoch die Aufstellung einige Fehler gehabt, die man so schnell nicht gefunden hätte.

3. Zum Lautwandel allgemein

Die wahrscheinlich bedeutendste Entdeckung in der historisch-vergleichenden Sprachwissenschaft ist das junggrammatische Postulat der *Ausnahmslosigkeit der Lautgesetze*. Im Vorwort zu den *Morphologischen Untersuchungen* schreiben Osthoff und Brugman (1878, S. XIII):

Die zwei wichtigsten von den methodischen Grundsätzen der ‘junggrammatischen’ Richtung sind folgende.

Erstens. Aller Lautwandel, so weit er mechanisch vor sich geht, vollzieht sich nach *ausnahmslosen Gesetzen*, d. h. die Richtung der Lautbewegung ist bei allen Angehörigen einer Sprachgenossenschaft, ausser dem Fall, dass Dialektspaltung eintritt, stets dieselbe, und alle Wörter, in denen der Lautbewegung unterworfenen Laut unter gleichen Verhältnissen erscheint, werden ohne Ausnahme von der Änderung ergriffen.

Zweitens. Da sich klar herausstellt, dass die Formassoziation, d. h. die Neubildung von Sprachformen auf dem Wege der Analogie, im Leben der *neueren* Sprachen eine sehr bedeutende Rolle spielt, so ist diese Art von Sprachneuerung unbedenklich auch für die *älteren* und *ältesten* Perioden anzuerkennen, und nicht nur überhaupt anzuerkennen, sondern es ist dieses Erklärungsprinzip auch in derselben Weise zu verwerten, wie zur Erklärung von Spracherscheinungen späterer Perioden, und es darf nicht im mindesten auffallen, wenn Analogiebildungen in den älteren und ältesten Sprachperioden *in demselben Umfange* oder gar *in noch grösserem Umfange* uns entgegentreten wie in den jüngeren und jüngsten.

Im darauf folgenden Absatz erklären sie, dass durch die Ausnahmslosigkeit der Lautgesetze man überhaupt erst “*festen Boden unter den Füßen*” bekomme. Dass Sprachwandel nicht allein aus ausnahmslosem Lautwandel besteht, liegt auf der Hand. Die Junggrammatiker stellen ihm die Formassoziation d.h. Analogie entgegen, aber auch die Kombination von ausnahmslosem Lautwandel und Analogie kann nicht allen Sprachwandel erklären: auch Dialektentlehnungen, expressivie Bildungen oder lexikalisch beschränkte Lautveränderungen wie Haplologie oder Dissimilationen können dem regulären Lautwandel entgegenwirken.

Im Kapitel “The regularity of sound change” deduziert Katičić (1970) auf Basis theoretischer Annahmen, dass regulärer Lautwandel ausnahmslos sein muss und alle scheinbaren Ausnahmen durch andere Sprachveränderungen erklärt werden müssen. Katičić unterscheidet externen und internen Sprachwandel. Beim externen Sprachwandel haben die Sprecher einer Sprache den Anreiz, eine neue Sprache zu lernen (z.B. die Sprache ihrer Eroberer). Beim internen Sprachwandel ist die neue Sprache innerhalb der Sprachgemeinschaft entstanden und die Sprecher haben prinzipiell keinen Anreiz, die Sprache der Innovatoren zu lernen. Wenn sich daher die neue Sprache durchsetzen will, muss sie mühelos von den Sprechern der alten Sprache gelernt werden können, was impliziert, dass die neue Sprache möglichst einfach aus der alten abgeleitet werden

können muss. Dies ist (auf die Lautebene bezogen) dann der Fall, wenn sich der Lautwandel als eindeutige Abbildung (d.h. als Funktion) von Lauten modellieren lässt, so dass ein Sprecher anhand möglichst einfacher Regeln aus jedem Laut der alten Sprache den entsprechenden Laut der neuen Sprache herleiten kann.

Dieser ausnahmslose und mechanische Lautwandel ist es, der von lga simuliert wird.

4. lga

4.1. Vorüberlegungen

Wenn ein Lautgesetz wirkt, wird ein Laut oder eine Lautgruppe durch eine andere oder durch nichts ersetzt. Da man Laute phonetisch notieren kann, kann man auch ein Lautgesetz durch Ersetzung von Zeichen durch andere Zeichen beschreiben.

Da Lautwandel meist durch die Lautumgebung bedingt ist und sich oft nicht nur ein Laut, sondern Lautklassen verändern (bspw. Stimmloswerdung von stimmhaften Lauten), ist es sinnvoll, eine Sprache zu entwickeln, die Lautgesetze kompakt beschreiben und von einem Computerprogramm verstanden werden kann. Eine naheliegende Wahl ist daher eine Form von *regulären Ausdrücken*, die in vielen Programmen implementiert und weithin bekannt sind. Reguläre Ausdrücke sind zwar für diesen Zweck nicht perfekt geeignet, aber für einen Prototypen ausreichend. Wenn sich gezeigt hat, welche Defizite reguläre Ausdrücke für diesen Anwendungsfall haben, wird man eine besser geeignete Sprache entwickeln und implementieren können.

Als Programmiersprache habe ich die Scheme-Implementation CHICKEN ¹ und die irregex-Engine für reguläre Ausdrücke ² gewählt.

4.2. Reguläre Ausdrücke

Hier eine (informelle) Beschreibung der hier verwendeten Untermenge regulärer Ausdrücke. Die vollständige Dokumentation für die irregex-Engine für CHICKEN findet sich unter dem oben genannten Link.

Die grundlegende Funktion von regulären Ausdrücken ist das Finden (engl. *match*) von Zeichenketten (String) anhand eines Musters. Der gematchte Text kann dann durch einen anderen ersetzt werden, wobei Teile des Matches wieder aufgenommen werden können.

Ein regulärer Ausdruck matcht einen String aus einem Alphabet. Dies ist hier die Menge der Unicode *Codepoints*, nicht die der Unicode *Grapheme*, welche nämlich aus mehreren Codepoints bestehen können.

¹<http://www.call-cc.org/>, abgerufen am 10.8.2015

²<http://wiki.call-cc.org/man/4/Unit%20irregex>, abgerufen am 10.8.2015

Ein Zeichen des Alphabets, das kein Metazeichen ist (dazu unten), matcht sich selbst. Metazeichen verlieren ihre spezielle Funktion, wenn ihnen ein Backslash \ vorangeht. Der Ausdruck `/foo/` matcht also den String "foo" (und auch das "foo" in "foobar").

Der Punkt `.` matcht ein beliebiges Zeichen. Der Ausdruck `/.../` matcht also alle Strings mit drei Zeichen, `/\.\.\./` matcht drei Punkte.

Eine in eckigen Klammern stehende Menge an Zeichen matcht eines dieser Zeichen, oder alle außer dieser Zeichen, wenn das erste Zeichen ein Zirkumflex `^` ist. Dabei können auch Zeichenbereiche mit Bindestrich angegeben werden. `/[abc]/` matcht also "a", "b" oder "c", `/[^abc]/` alle Zeichen außer "a", "b" und "c". Alternativ hätte man auch `/[a-c]/` bzw. `/^[a-c]/` schreiben können.

Der Zirkumflex `^` und das Dollarzeichen `$` matchen den Anfang bzw. das Ende einer Zeile. `/^foo/` matcht also "foo" am Anfang der Zeile, `/bar$/` matcht "bar" am Ende der Zeile und `/^quux$/` die Zeile, die nur "quux" enthält. Da in `lga` jedes Wort in einer eigenen Zeile steht, kann man die Zeichen für den Anfang bzw. das Ende des Wortes benutzen.

`?`, `*` und `+` sind Quantoren und matchen, was ihnen vorangeht, null oder einmal, null bis unendlich mal und ein bis unendlich mal. Der Ausdruck `/.[a-d]*x+/` z.B. matcht optional ein beliebiges Zeichen, dann eins der Zeichen "a"- "d" null bis unendlich mal und schließlich mindestens ein "x".

Mit Klammern `()` werden analog zur mathematischen Notation Matches enger gruppiert. So würde `/[ab]c+/` "a" oder "b" und dann mindestens ein "c" matchen (also z.B. "accc" oder "bc"). `/([ab]c)+/` dagegen matcht "[ab]c" mindestens einmal (also z.B. "acbcbc").

Neben ihrer Gruppierungsfunktion werden die Submatches innerhalb der Klammern für die Ersetzung gespeichert. `/([0-9]+)foo([0-9]+)/` würde also zwei Zahlen und "foo" dazwischen matchen. Die Zahlen wären dann in den Submatches 1 und 2 gespeichert und können im Ersetzungstext wieder aufgegriffen werden. Dabei gilt, dass verschachtelte Klammerausdrücke vor Klammern derselben Ebene numeriert werden. Also wären die Submatches in z.B. `/(a(b(c))(x))/` der Reihe nach "abc", "b", "c", "x".

Mit dem senkrechten Strich in z.B. `/e0|e1/` wird "e0" oder "e1" gematcht. Der Senkrechtrich hat die niedrigste Präzedenz, `/foo|bar+/` matcht also "foo" oder mindestens ein "bar". `/(foo|bar)+/` dagegen matcht mindestens ein "foo" oder "bar".

4.3. Funktionsweise

Im Kern arbeitet `lga` mit einer Liste von Wörtern und einer Liste von Lautgesetzen (in der Form von Textersetzungen durch reguläre Ausdrücke) und wendet der Reihe

nach alle Lautgesetze auf die Liste der Wörter an. Das Resultat sind die Wörter nach Anwendung aller Lautgesetze.

Die Liste der Lautgesetze aber hängt davon ab, von welcher in welche Sprache die Wörter transformiert werden sollen und wird aus einem *Baum* von Lautgesetzen und einem zugehörigen Sprachstammbaum generiert.

Die Liste der Wörter wird vorher durch eine weitere Liste von Regeln, die von der Ausgangssprache abhängt, von einer zugrundeliegenden graphematisch/phonologischen Form in eine interne phonetische gebracht (was natürlich eine eindeutige Graphie voraussetzt). Die Phoneme dieser Form sind als zugrundeliegende Segmente im Sinne der generativen, nicht der strukturalistischen Phonologie zu verstehen. Am Ende wird die phonetische Form durch eine Liste von Regeln, die von der Zielsprache abhängig ist, wieder in eine graphematische Darstellung gebracht.

Die Funktion (`run-list from to words`) generiert die drei Listen von Regeln anhand der Sprachen `from` und `to`, macht aus der Liste von Wörtern `words` aus Effizienzgründen einen einzigen String, in dem jedes Element von `words` in einer eigenen Zeile steht, wendet darauf die drei Regellisten an, und gibt das Ergebnis zurück.

Zunächst wird mit der Funktion (`make-path tree start end`) anhand eines Sprachstammbaums `tree` eine Liste aller Sprachen von der Ausgangs- zur Zielsprache generiert. Der Aufruf (`make-path lang-tree 'uridg 'lesb`) würde bspw. mit dem Beispielbaum unten die Liste (`uridg urgr nwgr aiol lesb`) erzeugen.

Der Sprachstammbaum (gespeichert in der globalen Variable `lang-tree`) definiert das Verwandtschaftsverhältnis von Sprachen. Wenn eine Sprache keine Untersprachen hat, ist sie (in Scheme-Terminologie) ein Symbol (also hat z.B. `myk` keine Abkömmlinge). Hat eine Sprache Untersprachen, ist sie eine Liste, dessen erstes Element ein Symbol für die Sprache selbst ist; die weiteren Elemente sind Untersprachen (wiederum Symbole oder Listen). Ein vereinfachter Stammbaum könnte also bspw. folgendermaßen aussehen:

```
(urgr (nwgr (dor nwdor sardor sdor)
          (aiol boiot (thess wthess othess) lesb))
      (sogr (ach myk ark kypr pamph)
            (ion-att ion att euboi)))
```

Es sei angemerkt, dass die so beschriebenen Verhältnisse nicht streng als genetische Verwandtschaft interpretiert oder zumindest so verwendet werden müssen. Auch nach einer Aufspaltung können nah verwandte Dialekte noch gemeinsame Entwicklungen haben. So passieren viele Lautveränderungen in allen griechischen Dialekten, sind aber dennoch nicht urgriechisch. Passiert also ein Lautwandel bspw. nach der Aufspaltung

von urion. in ion. und att. trotzdem noch in beiden Dialekten (dafür gibt es genug Beispiele), kann man ihn nicht als urion. bezeichnen. Für diese Fälle mag es sinnvoll sein, auf die Vorsilbe *ur-* zu verzichten oder z.B. statt von urion. von ion-att. zu sprechen (wie im Beispielbaum oben). Eine bessere Trennung von genetischer Verwandtschaft und “Sprachgruppen” wäre methodisch vielleicht angebracht. Die Praxis wird zeigen, ob hier noch Verbesserungsbedarf besteht.

Der Lautgesetzbaum ist eine Liste von vier möglichen Elementen, nämlich Funktionen, die einen String als Argument nehmen und einen String zurückgeben (also Funktionen, die Lautgesetze anwenden), Symbole, die eine erreichte Sprachstufe markieren, und zwei Arten von Listen des Typs (`br|sub lang-lst rule-tree`), deren zweites Element eine Liste von Sprachen ist, für die sie gelten (bzw. nicht gelten, wenn das erste Element `not` ist), und deren drittes Element ein weiterer Lautgesetzbaum ist. `br` (für *branch*) leitet einen neuen Sprachzweig ein, der durch `rule-tree` definiert ist (d.h. dass alle Regeln nach einem erfolgreichen `br` ignoriert werden). `sub` (für *subrules*) funktioniert ähnlich, macht aber nach Abarbeitung von `rule-tree` nach der Regel weiter. Dies impliziert, dass es sich bei `rule-tree` nicht um einen echten Baum handelt. `sub` ist nur dazu gedacht eine Liste von Lautgesetzen bedingt anzuwenden. Im Griechischen ist dies wie gerade erwähnt besonders nützlich, da zwar alle Dialekte ähnliche Entwicklungen machen, diese sich aber im einzelnen unterscheiden. Mit `sub` kann man einfach Regeln für bestimmte Dialekte oder Dialektgruppen definieren, ohne für jeden Dialekt einen eigenen Sprachzweig haben zu müssen, und so Sprachkontaktphänomene recht gut beschreiben.

Die Funktion (`make-rules tree path`) gibt anhand des Lautgesetzbaums `tree` eine Liste von Lautgesetzen zurück, die die Sprachentwicklung in `path` (von `make-path` erzeugt, s.o.) reflektiert. Von (`apply-rules rules words`) wird dann diese Liste `rules` auf die Wörter `words` angewandt.

Die Lautgesetzfunktionen werden von der Funktion (`s pattern . subst`) erzeugt, die aus einem String `pattern` einen regulären Ausdruck macht und eine Funktion zurückgibt, die `irregex-replace/all` auf diesen regulären Ausdruck und die Ersetzungen in der Liste `subst` anwendet. Matcht der reguläre Ausdruck einen Teil der Eingabe, wird diese durch `subst` ersetzt, dessen Elemente Strings, Zahlen als Indizes der Submatches, sowie Funktionen, die anhand eines Matches Strings zurückgeben, sind. (`irregex-replace/all "(foo)(bar)" "foobar" 2 (lambda (m) (string-reverse (irregex-match-substring m 1))) "quux"`) z.B. würde den String “foobar” durch “baroofquux” ersetzen.

Die oben beschriebenen regulären Ausdrücke sind für die Modellierung von Lautgesetzen nicht ausreichend. Neben anderen Unzulänglichkeiten, die sich im Laufe der

Arbeit herausstellen werden, gibt es eindeutig Probleme, mit Lautklassen umzugehen, und noch allgemeiner, Laute durch Zeichen zu kodieren.

Wenn jeder Laut durch genau ein Zeichen bzw. einen Unicode Codepoint kodiert wird, gibt es bei der Lautkodierung keine Probleme. Da es allerdings sinnvoll ist, einige Laute mit mehreren Codepoints zu kodieren, kann es zu Problemen kommen. Will man bspw. ein Lautgesetz (s "b" "p") formulieren, so hätte dies die Nebenwirkung, dass auch "b^h" zu "p^h" würde, was im allgemeinen Fall unerwünscht ist. Wenn man Laute wie *b^b* oder *g^w* mit einem einzigen Codepoint repräsentieren würde, hätte man dieses Problem natürlich zwar nicht, jedoch bietet Unicode für solche Vorhaben keine definierten Codepoints. Dafür müsste man die Private Use Area benutzen, wodurch die Kodierung jedoch von der Schriftart abhängig wird, eine unschöne Lösung. Eine Möglichkeit, dieses Problem zu umgehen, ist, Polygraphen sowohl in den zu transformierenden Wörtern als auch in den die Lautgesetze beschreibenden regulären Ausdrücken durch beliebige unbenutzte Codepoints zu ersetzen und am Ende die umgekehrte Ersetzung durchzuführen, so dass die nur intern verwendeten Codepoints nirgendwo graphisch auftauchen. Tatsächlich benutze ich auch in den regulären Ausdrücken in einigen Fällen eine einfachere Kodierung (wenn eine gute Alternative vorhanden war, z.B. "w" statt "u") wie in 5.1 beschrieben (die Ersetzung der komplexen durch eine einfachere Kodierung in den Wörtern findet in der Liste der phonologischen Regeln statt, der umgekehrte Fall in den graphematischen). In diesen Fällen ist jedoch die einfache Kodierung im regulären Ausdruck direkt verwendet – also keine Ersetzung von komplexer durch einfache Kodierung – aber eine automatische Ersetzung könnte in (make-regex s) leicht implementiert werden. Der Einfachheit halber und aus Effizienzgründen habe ich jedoch zunächst darauf verzichtet, da die problematischen Fälle wenige sind und vorläufig durch vorsichtige Formulierung umgangen werden können (der obere Fall also als (s "b([^h])" "p" 1)).

Bei der Formulierung von Lautgesetzen ist es ferner sinnvoll, Laute in Lautklassen zusammenzufassen, um eine konzise und leicht abstrahierende Darstellung zu haben. Der POSIX-Standard definiert zwar Zeichenklassen wie [:digit:] für Ziffern oder [:lower:] für Kleinbuchstaben, aber die Bedürfnisse, die man als Linguist hat, sind damit nicht abgedeckt, da sich zur Laufzeit Zeichenklassen weder neu definieren noch verändern lassen. Dieses Problem lässt sich vorläufig lösen, indem die regulären Ausdrücke, bevor sie von string->irregex in eine interne Form kompiliert werden, noch verändert werden, so dass die tatsächlich im Quelltext vorkommenden regulären Ausdrücke erst zu echten regulären Ausdrücken verarbeitet werden, die von der irregex-Engine verstanden werden. Konkret funktioniert dies so, dass in einer assoziativen Liste Lautklassennamen (als Symbole) mit Strings assoziiert werden und bei

der Vorbearbeitung der regulären Ausdrücke Strings der Form `<lautklassenname>` durch den mit dem Symbol `lautklassenname` assoziierten String ersetzt werden. Die Funktion `(set-class key value)` assoziiert den Namen `key` mit dem String `value` und `(l key)` (kurz für *lookup*) findet den zu `key` passenden String in der assoziativen Liste.

So kann man bspw. mit `(set-class 'kurz-vok "a|e|i|o|u")` und `(set-class 'lang-vok "ā|ē|ī|ō|ū")` Kurz- und Langvokale definieren, mit `(set-class 'vok (s+ (l 'kurz-vok) "|" (l 'lang-vok)))` dann die Menge aller Vokale (`s+` ist eine Abkürzung für `string-append` zum Verketteten von Strings) und diese in einem Lautgesetz wie `(s "<vok>" "a")` verwenden.

Die Lautklassen sind jedoch im Laufe der Sprachentwicklung nicht vernünftig veränderbar, da sie zur Zeit der Auswertung von `s` in den regulären Ausdruck eingefügt werden (also wenn die Quelltextdatei gelesen wird), man die Definition aber wohl am ehesten aus dem Lautgesetzbaum heraus verändern wollen würde (Lautklassen sollten also dynamisch statt lexikalisch gebunden sein). Hier wird nur eine neuentwickelte Lautgesetzengine wirklich Abhilfe schaffen können.

Es gibt jedoch noch ein weiteres Problem. So kann man nicht ohne weiteres z.B. einen Langvokal durch seinen entsprechenden Kurzvokal oder einen silbischen durch sein unsilbisches Pendant ersetzen. Beispiele für solche Vorgänge gibt es genug und es ist eine Methode notwendig, mit der man diese Fälle unkompliziert ausdrücken kann. In einer späteren Version wäre vielleicht eine Ersetzung wie `(s "<lang-vok>" "<kurz-vok>")` wünschenswert. In der aktuellen Version können jedoch auch Funktionen, die auf Submatches angewandt werden, denselben Effekt erzielen. Als Ersetzungsargumente kann man an `irregex-replace/all` neben Strings und Zahlen, die das entsprechende Submatch bezeichnen, auch Funktionen übergeben, die auf das aktuelle Match angewandt werden und einen String zurückgeben. Um also z.B. einen Langvokal in ein Kurzvokal zu verwandeln, braucht man nur eine Funktion, die auf ein Submatch eine Reihe von regulären Ausdrücken anwendet und das Ergebnis zurückgibt. Eine Funktion, die reguläre Ausdrücke auf einen String anwendet, gibt es ja schon: `apply-rules`. Diese wird von der Funktion `(match-rulelist rules)` benutzt, die eine Liste von Ersetzungsfunktionen `rules` auf ein Submatch anwendet. Dazu gibt sie eine Funktion zurück, die einen Submatchindex `i` bindet und eine Funktion zurückgibt, auf die `irregex-replace/all` angewandt werden kann. Die Vokalkürzung könnte somit folgendermaßen definiert werden:

```
(define kuerzung
  (match-rulelist
    (list (s "ā" "a"))
```

```
(s "ē" "e")
(s "ō" "o")
(s "ī" "i")
(s "ū" "u"))))
```

Ein Lautgesetz, das alle Langvokale kürzt, kann dann als (s "(<lang-vok>)" (kuerzung 1)) formuliert werden.

Die schon erwähnten Regeln zur Umwandlung von einer graphematisch/phonologischen in eine phonetische und von einer phonetischen in eine graphematisch/phonologische Darstellung sind unter ihrem Sprachnamen ebenfalls in assoziativen Listen abgelegt.

Hiermit ist die Funktionalität von lga weitestgehend erläutert. Eine exakte Beschreibung kann natürlich nur der Quelltext selbst liefern.

5. Modellierung der Lautentwicklung des Altgriechischen

5.1. Vorbemerkungen

Vor der Beschreibung der Lautentwicklung noch einige Anmerkungen, die nicht speziell das Griechische betreffen. Aufgrund von Fehlern in der Implementation von irregex können in regulären Ausdrücken keine höheren Unicodezeichen in []-Sets benutzt werden. Da diese aber notwendig sind, um gewisse Zeichen auszuschließen – insbesondere ^h zur Unterscheidung von Aspiraten – müssen diese aus dem ASCII-Vorrat stammen. Aus dem Grund wird für die Aspiration ! statt ^h verwendet.

Für einige komplex kodierte Laute (d.h. mehr als ein Codepoint) wird wie schon beschrieben in den regulären Ausdrücken eine einfachere Kodierung benutzt. Für die Labiovelare wurden im Gegensatz zu den Velaren Großbuchstaben verwendet (also z.B. *K* statt *k^w*), die silbischen Resonanten *ŕl̥m̥n̥* werden ebenfalls mit den Großbuchstaben RLMN ausgedrückt. Die Halbvokale *i̯u̯* werden mit *yw* und die Laryngale *h*_[123] mit *H*_[123] bezeichnet. Akzente werden in der graphematischen Schreibweise als combining diacritics geschrieben und intern mit ' und ~ kodiert.

Als phonologische Regeln für das Uridg. sind Laryngalumfärbung (**e* neben **h₃* jedoch noch von **o* verschieden, was allerdings für das Gr. belanglos ist) und Stimmhaftwerdung von **s* neben stimmhaften Konsonanten beschrieben. Weitere phonologische Regeln müssen ggf. ergänzt werden. Lautgesetze wie STANGS Gesetz, die vor dem Erreichen des Sprachzustandes, der mit uridg bezeichnet wird, durchlaufen wurden, wurden nicht berücksichtigt und die Rekonstrukte bzw. Transponate, die als Eingabe dienen, sollten einen entsprechenden Lautstand vorweisen (um bei STANGS Gesetz zu

bleiben wäre also **dǣm* nicht **dǣum* uridg). Insbesondere enthält uridg den *þ*-Laut, dessen Herkunft und Phonetik hier ebenfalls nicht weiter hinterfragt wird.

Im Folgenden werden die Lautgesetze vom Urindogermanischen ins Griechische in drei Etappen aufgestellt. Zunächst wird die Lautentwicklung bis ins Urgriechische modelliert, dann bis in mykenischer Zeit und schließlich in die Dialekte der alphabetischen Zeit. Die dazu verwendete Literatur ist im wesentlichen Bartoněk 1966, Bartoněk 1991, Bartoněk 2003, Buck 1910, Lejeune 1972, Rix 1992 und Sihler 1995.

Eine (relative) Chronologie aller Lautgesetze lässt sich nicht immer mit Sicherheit aufstellen, da viele Lautgesetze kaum mit anderen Lautgesetzen interagieren oder mehrere Entwicklungen denkbar sind, so dass man gerade in der frühesten Zeit, in der das Griechische noch nicht belegt ist, in diesen Fällen wenig mehr als nur raten kann. Hier werden dann die Lautgesetze entweder mit ähnlichen anderen Gesetzen oder nach Gefühl einsortiert.

Auch ist die Aufstellung hier keinesfalls als endgültig anzusehen. Besonders Entwicklungen, die in die spätere dialektale Zeit fallen wie Digammaschwund und Kontraktionen, die zum Teil nach Einsetzen der Überlieferung stattfinden, sind zu unübersichtlich und unklar, um sie hier angemessen zu modellieren ohne den Rahmen der Arbeit zu sprengen (Schwierigkeiten werden im Text genannt). Aber auch über sicherere Entwicklungen lässt sich im Detail fast immer streiten. Sowohl das Programm *lga* als auch die Formulierung der Lautgesetze sind somit als vorläufiges Ergebnis oder Gerüst zu sehen, auf dem weitere Arbeiten aufbauen können. Insbesondere sind Akzententwicklungen noch völlig unbeachtet geblieben.

Nach der Beschreibung und Diskussion der Lautgesetze folgt eine formelhafte Darstellung, die mehr oder weniger genau den Regeln in *lga* entspricht. Zu jeder Regel werden einige Beispiele genannt, die zum Großteil von Rix übernommen, allerdings nicht erschöpflich sind. Sie dienen mehr zur Veranschaulichung denn als Belege. In besonders trivialen Fällen gebe ich daher auch weniger oder gar keine Beispiele. Auch in sehr unübersichtlichen Fällen bilden die Beispiele nicht die ganze Komplexität ab.

Sofern bei Rekonstrukten keine Sprachbezeichnung angegeben ist, ist irgendeine nicht weiter spezifizierte Vorform gemeint. Die Sternformen sind auch nicht in allen Fällen als Rekonstrukte zu verstehen. Wenn eine Form als uridg. bezeichnet ist, kann sich dies auch nur auf die einzelnen Morphe beziehen, ohne dass damit gesagt würde, dass schon das Uridg. die Form in dieser Zusammensetzung besessen hat.

5.2. Das Lautsystem des Urindogermanischen

Das hier angenommene Lautsystem des Uridg. sieht wie folgt aus. Umstrittene Laute stehen in Klammern. Teilweise ist nur der Phonemstatus umstritten, aber obwohl Pho-

neme für Lautgesetze eigentlich uninteressant sind, sind die hier aufgeführten Laute natürlich immer noch eine Abstraktion über die tatsächlichen Laute, was aber in der Natur der Rekonstruktion liegt und auch aus pragmatischen Gründen kaum anders gehandhabt werden könnte.

Vokale: **a*, **ā*, **e*, **ē*, **o*, **ō*, **i*, (**ī*), **u*, (**ū*)

Silbische Resonanten: **r*, **l*, **m*, **n*

Resonanten und Halbvokale: **r*, **l*, **m*, **n*; **i̯*, **u̯*

Verschlusslaute:

Labiovelare **k^w* **g^w* **g^{wb}* (**k^{wb}*)

(Rein)velare **k* **g* **g^b* (**k^b*)

Palatale **k̂* **ġ* **ġ^b* (**k̂^b*)

Dentale **t* **d* **d^b* (**t^b*)

Labiale **p* **b* **b^b* (**p^b*)

Frikative und sonstige: **s*, **z*, **h₁*, **h₂*, **h₃*, (**ʃ*)

**z* ist Allophon von **s* vor stimmhaften Konsonanten. Die beiden Laute werden der Einfachheit halber jedoch in den Regeln und in lga unterschiedslos mit **s* bezeichnet.

Die genaue Artikulation der Laryngale **h₁*, **h₂*, **h₃* ist nicht klar. Meist werden Frikative vermutet, neben denen ggf. Sprossvokale eingefügt werden.

Der Laut **ʃ* ist umstritten, sowohl seine Existenz als auch seine Artikulation (vielleicht dentale Affrikata?).

5.3. Urindogermanisch bis Urgriechisch

Die Lautveränderung, die ins Ugr. führen, sind grob gesprochen die Kentumvertretung der Tektale, die Entwicklung von **ʃ*, die Stimmloswerdung der Mediae aspiratae, jegliche Laryngalentwicklungen, die Entwicklung der silbischen Resonanten (teilweise aber auch erst dialektal), einige Okklusivassimilationen, die Entwicklung von **s* und die Anfänge diverser Palatalisierungen.

Da in dieser Phase die Lautveränderungen weniger ineinander greifen, ist eine relative Chronologie oft nicht aufzustellen.

1 Kentum Das Griechische ist eine Kentumsprache, es fallen also die uridg. Velare und Palatale zusammen (Rix §92-94). Da das Griechische keinerlei Anhaltspunkte dafür bietet, dass es je einen Unterschied zwischen diesen Lauten gab, wird der Zusammenfall früh datiert.

k̂, ġ, ġ^b, (k̂^b) > k, g, g^b, (k^b)

uridg. **k̂*: **k̂mtóm* > gr. ἑκατόν; ai. *śatám* 'hundert'

uridg. **k*: **leukós* > gr. λευκός; ai. *rocás* ‘leuchtend’
 uridg. **ġ*: **h₂éġonti* > att. ἄγουσι; av. *azənti* ‘sie führen’
 uridg. **g*: **gerh₂nós* > gr. γερανός; cf. lit. *gervė* ‘Kranich’
 uridg. **ġ^b*: zu **ueġ^b-/uoġ^b-*: att. ὄχος ‘Wagen’; av. *vazāmi* ‘ich fahre’
 uridg. **g^b*: **h₃mig^b-/h₃moig^b-*: gr. ὀμίχλη ‘Nebel’; ai. *meghás* ‘Wolke’

2 Thorn **p* (genauer Lautwert unklar) tritt nach Velaren und Labiovelaren auf. Nach Tenuis entspricht **t*, nach Media aspirata **d^b* (Rix §81).

p > *t* \ {Tenuis} _

uridg. **k̂p*: **ték̂pō(n)* > gr. τέκτων; ai. *tákṣā* ‘Zimmermann’

p > *d^b* \ {Media asp.} _

uridg. **g^{wb}p*: **h₃g^{wb}pitom* > gr. ἄφθιτον; ai. *ákṣitam* ‘unvergänglich’

3 Media aspirata > Tenuis aspirata Mediae aspiratae werden zu Tenuis aspiratae (Rix §94). Aufgrund der Nähe zum Makedonischen und Phrygischen, in denen der Reflex vermutlich Media ist, ist dieser Wandel vielleicht später anzusetzen. Spätestens im Myk. ist die Stimmlosigkeit belegt.

b^b, d^b, g^b, g^{wb} > *p^b, t^b, k^b, k^{bw}*

uridg. **b^b*: **b^béronti* > dor. φέροντι; ai. *bháranti* ‘sie tragen’

uridg. **d^b*: zu **d^beb₁-*: gr. τίθημι; ai. *dadhámi* ‘ich setze/stelle/lege’

uridg. **g^b*: zu **steiġ^b-*: gr. στείχω; got. *steigan* ‘steigen’

uridg. **g^{wb}*: zu **g^{wb}en-*: gr. φόνος ‘Mord’; ai. *g^bnánti* ‘sie erschlagen’

4 *(H)*i*- Im Gr. hat **i*- scheinbar zwei Reflexe, nämlich Verschärfung zu einem Laut, der mit **d_i*- und **g_i*- zusammenfällt und einfaches **i*-, das im Myk. noch belegt ist. Mit der Laryngaltheorie versucht man die beiden Reflexe als **H_i*- und **i*- zu unterscheiden. Welcher Anlaut verschärft wurde, ist unklar – nach Rix §68, 80e ist es **H_i*- – und nach dem Datenmaterial kann für beides argumentiert werden, wenn man aber ὑγιής ‘gesund’ aus **h₂i_u-g^{wh}ih₃-és* ‘langes Leben habend’ herleiten will, darf man **H_i*- > **i*- vermuten (**su*- ‘gut’, das Rix (§97) rekonstruiert und mit ai. *su*- ‘gut’ verbindet, ist als **h₁su*- anzusetzen und würde εὖ- ergeben). Auch lautlich plausibel ist ein früher Zusammenfall von uridg. **i*- und **d_i*- (cf. lat. *Iūlius* > ital. *Giulio* und *diŭrnus* > *giorno* ‘Tag’), danach Wegfall aller Laryngale (also keine Vokalisierung wie vor **u*), wodurch **H_i*- als **i*- stehenbleibt, wie es im Frühmyk. belegt ist.

i- > *d_i*-

uridg. **i*:- **iugóm* > gr. ζυγόν; ai. *yugám* ‘Joch’

$H\dot{i}- > \dot{i}-$

uridg. $*h_2\dot{i}-$: ὕγιής ist mit $*h_2\dot{i}u-$ ‘langes Leben (\rightarrow lange Zeit \rightarrow Zeit)’ wegen Vollstufe $*h_2e\dot{i}u-$ in z.B. hom. αἰεῖ ‘immer’ < $*h_2e\dot{i}u\epsilon si$ ‘zu allen Zeiten’ zu rekonstruieren.

5 Laryngale Die Entwicklung der Laryngale folgt Rix §79-85. Teilweise sind die Lautgesetze weniger explizit als bei Rix formuliert, um eine lautlich plausible Entwicklung zu modellieren. So wird z.B. intervokalischer Laryngal nach $*i$ oder $*u$ nicht zum Gleitlaut sondern dieser entsteht erst als Hiattilger nach dem Laryngalschwund. Entgegen Rix glaube ich nicht an Vokalisierung im Anlaut vor $*u$ (§79b), da die Beispiele nicht überzeugen und er immerhin selber ὕδῆω als Gegenbeispiel nennt. Der Schwund der Laryngale dürfte früh passiert sein. Der Wandel, der hier als $H > V$ geschrieben ist, ist als $h_1, h_2, h_3 > e, a, o$ zu verstehen.

$H > V \setminus C_ \#$ (Rix §85d)

uridg. $*-h_1 > \text{gr. } -\epsilon \text{ N.Du.m.}$

uridg. $*-h_2 > \text{gr. } -\alpha \text{ N.Pl.n.}$

$(i, u)H > (i, u)V \setminus CC_ \#$ (Rix §85b)

uridg. $*tr\acute{i}-h_2 > \text{gr. } \tau\rho\acute{\iota}\alpha; \text{ ai. } tr\acute{i}$ ‘drei’

$(i, u)H > (\dot{i}, \dot{u})V \setminus C_ \#$ (Rix §85b)

uridg. $*h_3\acute{o}k^w-ih_1 > *ok^w\acute{i}e > \text{hom. } \acute{o}\sigma\sigma\epsilon$ ‘beide Augen’

uridg. $*d\acute{o}ru-h_2 > *d\acute{o}r\acute{u}a > \text{hom. } \delta\omicron\upsilon\rho\alpha$ ‘Speere’

$VH > \bar{V} \setminus _ (C, R, \#)$ (mit $R_\circ > R$) (Rix §82b, 85a)

uridg. $*eh_1$: zu $*d^beh_1-$: gr. τίθημι ‘setze/stelle/lege’

uridg. $*eh_2$: zu $*steh_2-$: dor. ἵστανμι ‘stehe’

uridg. $*eh_3$: zu $*deh_3-$: gr. δίδωμι ‘gebe’

uridg. $*o/eh_2$: zu $*b^boh_2-/ *b^beh_2-$: dor. φωνά ‘Stimme’ bzw. φᾶτι ‘spricht’

uridg. $*-eh_2$ (N.Sg. eh_2 -St.) > ion-att. $-\eta$, sonst $-\bar{\alpha}$; ai. $-\bar{a}$

uridg. $*-oh_1$ (N.Du. o -St.) > $-\omega$; ai. $-\bar{a}$

$H > V \setminus (\#, C)_ C$ (Rix §80d, 82c)

uridg. $*h_2u-$: $*h_2ueh_1ti > \text{hom. } \acute{\alpha}\eta\sigma\iota; \text{ ai. } v\acute{a}ti$ ‘weht’

uridg. $*h_1r-$: $*h_1reg^wos > \text{gr. } \acute{\epsilon}\rho\epsilon\beta\omicron\varsigma$ ‘Finsternis’; ai. $r\acute{a}jas$ ‘Dunst’

uridg. $*d^bh_1-t\acute{o}s > \text{gr. } \theta\epsilon\tau\acute{o}s$ ‘gesetzt/-stellt/-legt’

uridg. $*ph_2t\acute{e}r > \text{gr. } \pi\alpha\tau\acute{\eta}\rho; \text{ ai. } pit\acute{a}$ ‘Vater’

uridg. $*dh_3-t\acute{o}s > \text{gr. } \delta\omicron\tau\acute{o}s$ ‘gegeben’

$\eta h_2 > na \setminus _ \#$ (Rix §85c)

uridg. $*\acute{\epsilon}h_2h_2\eta h_2$ (N.Pl.n.) > hom. κάρηνα; ai. $\acute{s}ir\acute{s}\acute{a}$ ‘Köpfe’

$HR_2 > VR \setminus C_-$ (Rix §84γ)

uridg. $*h_1\eta$: zu $*d^{bid}h_1-\eta t-$: gr. τιθέ-ντ-ος; ai. *dádh-at-as*

uridg. $*h_3\eta$: zu $*didh_3-\eta t-$: gr. διδό-ντ-ος; ai. *dád-at-as*

$RH > R\bar{V} \setminus _C$

uridg. $*l h_1$: zu $*g^w l h_1-$: gr. βλητός ‘geworfen’; ai. *ud-gūr-ŋas* ‘emporgehoben’

uridg. $*l h_2$: $*t l h_2 tós >$ dor. τλᾱτός; lat. *lātus* ‘getragen’

uridg. $*g h_3$: $*st g h_3 tós >$ gr. στρωτός; lat. *strātus* ‘hingebreitet’

$\acute{R}H > \acute{V}RV \setminus _C$

uridg. $*h_1$: $*g h_1 tis >$ gr. γένεσις ‘Erzeugung’

uridg. $*f h_2$: $*k f h_2 s \eta h_2 > *k árasna >$ hom. κόρηνα ‘Köpfe’

uridg. $*f h_3$: zu $*ste-st f h_3-$: aiol. ἐ-στόρο-ται

$HR_2 > VR \setminus \#_-$ (Rix §79c)

uridg. $*h_1 r-$: zu $*h_1 r-ske-$: gr. ἔρχεται; ai. *rcháti* ‘kommt’

uridg. $*h_3 \eta-$: zu $*h_3 \eta b h_1-/*h_3 nob h_1-$: gr. ὀμφαλός, ahd. *nabulo* ‘Nabel’

$H > \emptyset$

6 $*ue\grave{u} > ue\grave{i}$ $*ue\grave{u}$ wird zu $*ue\grave{i}$ dissimiliert, wie εἶπον < $*e-ue-u k^w-om$ (vgl. ai. *avocam*) lehrt. Dies muss ferner vor 7 passieren, wie εἶπον ebenfalls zeigt.

$u > i \setminus ue_-$

7 Labiovelar > Velar Neben $*u$, $*u$ und vor $*i$ fallen die Labiovelare mit den Velaren zusammen (Rix §97).

$k^w, g^w, k^{wb} > k, g, k^b \setminus (u, \bar{u}, u_-); _-(u, \bar{u}, u, i)$

uridg. $*u k^w$: zu $*-k^w ólos$: gr. βουκόλος ‘Kuhhirte’ vs. αἰπόλος ‘Ziegenhirte’ (zu Labiovelar > Labial s. 32)

uridg. $*u g^w$: zu $*-g^w i h_3-$: gr. ὑγίης ‘gesund’ vs. βίος ‘Leben’

uridg. $*g^w b u$: zu $*h_1 l \eta g^w b-$: gr. ἐλαχύς vs. ἐλαφρός ‘schnell’

uridg. $*k^w i$: zu $*h_3 o k^w-$: hom. ὅσσε < $*o k^w-je$ ‘beide Augen’; lit. akis ‘Auge’

8 Silbische Resonanten Die Entwicklung der silbischen Resonanten folgt Rix §75 und §76. Für das Urgr. wird ein Sprossvokal $*ə$ angesetzt, welcher sich dialektal zu *a* oder *o* entwickelt (s. 27).

Hierunter fallen auch die Fälle, wo ursprünglich Laryngal zwischen Resonant und Vokal stand. Diese sind bei Rix gesondert behandelt, was aber unnötig ist, wenn man den Laryngalschwund einfach vorher ansetzt.

Genauere Datierung unklar. Nach Laryngalschwund. Eine etwas spätere Datierung wäre auch möglich, aber vor 21 wegen z.B. $*g^w am i \bar{o} > βαίνω$.

$N_{\circ} > aN \setminus (V, \check{V})$

* $\check{m}V$: zu uridg. * $\check{k}mh_2$ –: gr. ἔκαμον ‘mühte sich’

* $\check{n}i$: uridg. * $tek\check{h}nih_2$ > * $tektan\check{i}a$ > gr. τέκταινα ‘Erbauerin’

$N_{\circ} > \emptyset$

* \check{n} : uridg. * $t\check{n}t\acute{o}s$ > gr. τατός; ai. tatás, lat. tentus ‘gespannt’

* \check{m} : uridg. * $d\acute{e}k\check{m}$ > att. δέκα, ark. δέκο; lat. decem ‘zehn’

$L_{\circ} > \emptyset L_{\circ} \setminus (\#, V, \check{V})$

uridg. * $H\check{i}\acute{e}k\check{w}_r$ > gr. ἥπαρ ‘Leber’

* $ekstr\check{i}\acute{o}$ > * $ekstar\check{i}\acute{o}$ > gr. ἐχθαίρω ‘hasse (den Feind)’

zu uridg. * $g^w\check{h}_1$ –: * $eg^w\check{a}lon$ > gr. ἔβαλον ‘sie warfen’

$L_{\circ} > L\emptyset$

uridg. * $str\acute{o}s$ > att. στρατός, lesb. boiot. στροτός ‘Truppe, Heer’

uridg. * $plth\acute{u}s$ > att. πλατύς; ai. prthús ‘breit’

9 Nasal vor Okklusiv Vor Okklusiven wird der homorgane Nasal realisiert (Rix §78). Gegen Rix jedoch nicht * ms > * ns wegen ἔνειμα < * $enemsa$. Hier kann m zwar analog wieder eingeführt worden sein, aber ohne ein Beispiel, das bei Rix fehlt, ist die Regel zunächst unnötig. Die Nasalassimilation passiert auch später noch und dürfte tatsächlich wohl über lange Zeit eine synchrone phonologische Regel sein.

$N > m \setminus \{\text{Labial}\}$

$N > n \setminus \{\text{Dental, Velar, Labiovelar}\}$

10 * $m\check{i}$ > * $n\check{i}$ Rix §77. Datierung unklar. Vielleicht nach * \check{m} > * am , aber * \check{m} > * $\check{n} \setminus \check{i}$ ist ebenfalls als zusätzliche Regel denkbar. Vielleicht auch im Zuge der Palatalisierung (s. 21).

$m > n \setminus \check{i}$

zu uridg. * g^wem –: * $g^wm\check{i}\acute{o}$ > * $g^wam\check{i}\acute{o}$ > * $g^wan\check{i}\acute{o}$ > gr. βαίνω ‘ich komme’; lat. veniō

11 Benachbarte Vokale nach Laryngalschwund Kontraktion von * e , * a , * o , zwischen denen Laryngal geschwunden ist (Rix §81). Hier entsteht zuerst der Intonationsunterschied zwischen Akut und Zirkumflex. Trägt der erste Vokal den Akzent, ist das Kontraktionsprodukt zirkumflektiert. Trägt ihn der zweite, ist es akutiert. Kontrahierte * ee , * oo fallen mit den alten (offenen) Langvokalen zusammen; die geschlossenen Langvokale entstehen erst bei 40. Datierung unklar.

$VV > \bar{V}$ ($V = a, e, o$)

uridg. * $eh_2e\check{i}$ D.Sg. eh_2 -St. > * $aa\check{i}$ > ion-att. -η, sonst -α

Nach Vokal werden **i*, **u* zu Halbvokalen (es entstehen also Diphthonge).

Im umgekehrten Fall entsteht ein Gleitlaut zwischen den beiden Lauten. Dieser ist streng genommen nicht gesichert, da er nicht geschrieben wird, da aber die Silbengrenze erhalten bleibt, ist er phonetisch äußerst wahrscheinlich (und sollte vielleicht vordatiert werden).

$i, u > i, u \setminus V_$

uridg. **pléh₁istom* > gr. πλεῖστον 'am meisten'

$i, u > i, u \setminus _V$

uridg. **tri-h₂* > gr. τρία 'drei'

zu uridg. **h₃b^hruH-*: gr. ὄφρυος 'Braue'

12 Okklusivassimilation Okklusive werden an folgende Okklusive in Aspiration und Stimmhaftigkeit assimiliert (Rix §106a).

Hier dürfte es sich wohl eigentlich um eine phonologische Regel handeln.

{Okklusiv} > {Media} \ {Media}

gr. ἐμπλέγ-δην zu ἐμπλέκω

gr. κρύβ-δην zu κρύφα

{Okklusiv} > {Tenuis} \ {Tenuis}

gr. ἐκ-τός zu ἔχω

gr. ληπ-τός zu λαμβάνω

{Okklusiv} > {Aspirata} \ {Aspirata}

gr. ἐδιώχ-θην zu διώκω

gr. ἐλήφ-θην zu λαμβάνω

13 **K_u* > **K^w*(:) Velar + **u* wird zum Labiovelar, zwischen Vokalen als Geminat (Rix §104), um das Silbengewicht zu erhalten (vielleicht nur nach Kurzvokal?).

$K_u > K^w \setminus V_V$

uridg. **h₁ék_uos* > **ék^wk^wos* » gr. ἵππος (mit rätselhaftem Anlaut); ai. *ásvas* 'Pferd'

$K_u > K^w$

uridg. zu **ǵ^hueh₁r-*: hom. θήρ, aiol. φήρ; lit. žvėris 'wildes Tier'

14 Beseitigung von Geminaten Das Gr. muss eine Zeit lang Geminaten vermieden haben. Entweder generell wie das Uridg. – dann müsste diese Regel vor 13 kommen und dort die Geminatbildung nur nach Kurzvokal passieren – oder nur im Silbenanlaut (also nur nicht nach Kurzvokal) wie αἰπόλος < **a₁k^wolos* < **aig-k^wolos* zeigt. Weitere Beispiele könnten die Verhältnisse klarer machen.

Datierung unklar, vermutlich nach 13.

$C: > C \setminus _$

$*aig-k^w\acute{o}los > *aik^wk^w\acute{o}los > *aik^w\acute{o}los > \text{gr. } \alpha\iota\pi\acute{o}\lambda\omicron\varsigma \text{ 'Ziegenhirte' zu gr. } \alpha\acute{\iota}\xi, \alpha\iota\gamma\acute{o}\varsigma \text{ 'Ziege'}$

15 Okklusive neben *s

Vor *s werden alle Okklusive zu Tenuis, wobei gleichzeitig Stimmhaftigkeit und Aspiration auf einen evtl. folgenden Okklusiv übertragen werden (Rix §105).

Da die Formulierung etwas komplex ist, modelliere ich es in zwei Schritten: zuerst Assimilation, dann Tenuiswerdung.

Vor 16 zu datieren.

$\{\text{Tenuis}\} > \{\text{Media}\} \setminus \{\text{Media}\}_s _$

$*mig-sk\acute{o} > *migs\acute{o} > *miks\acute{o} > \text{gr. } \mu\acute{\iota}\sigma\gamma\omega \text{ 'ich vermische'}$

$\{\text{Tenuis}\} > \{\text{Aspirata}\} \setminus \{\text{Aspirata}\}_s _$

$*p^{bat^b}-sk\acute{o} > *p^{bat^b}sk^b\acute{o} > *p^{batsk^b}\acute{o} > \text{gr. } \pi\acute{\alpha}\sigma\chi\omega \text{ 'ich erleide'}$

$\{\text{Okklusiv}\} > \{\text{Tenuis}\} \setminus _s$

s.o.

$*aig-si > \alpha\iota\acute{\iota}\varsigma\acute{\iota} \text{ 'den Ziegen'}$

$*t^{brik^b}-si > \theta\rho\acute{\iota}\xi\acute{\iota} \text{ 'den Haaren'}$

16 CsC > sC Steht *s zwischen zwei gleichen Konsonanten, schwindet der erste (Rix §104). Dabei werden Stimmhaftigkeit und Aspiration ignoriert, da vor *s nach 15 nur Tenuis stehen kann.

$C_1 > \emptyset \setminus _C_2 \ (C_1 = C_2)$

s. 15

$*dikskos > \delta\acute{\iota}\sigma\kappa\omicron\varsigma \text{ 'Scheibe'}$

Urgriechisch I

Das hier erreichte Sprachstadium ist eine Vorstufe des Urgriechischen, das als Ausgangssprache bei der Lautwandelsimulation sehr nützlich ist, da man nicht alle Vorformen auf das Urindogermanische zurückführen will, das Urgriechische andererseits in der Entwicklung oftmals schon zu weit fortgeschritten ist. Was Rix als urgr. bezeichnet, entspricht oftmals eher meinem urgr. I, manchmal sogar noch früheren Sprachstufen.

17 *s > *b *s wird nach Vokal oder im Anlaut und vor Vokal, Resonant oder *u, nach unbetonter Silbe zwischen Resonant oder *u und Vokal und zwischen Nasal und Vokal zu *b verhaucht (Rix §86, 89). Dieses *b schwindet in fast allen Fällen später ganz (teilweise mit Ersatzdehnung oder Geminat, s. 40, 43).

In einigen Fällen ist *s auf unerklärte Weise erhalten, teilweise beim selben Wort: σῶς neben ὄς ‘Schwein’, σμικρός neben μικρός ‘klein’.

Der Wandel ist vor OSTHOFF (19) zu datieren.

$s > h \setminus (\#, V)_{-}(V, R, \text{u})$

*#sV: uridg. *septm̥ > gr. ἑπτὰ; ai. *saptá* ‘sieben’

*#sr: zu uridg. *sreṃ-: gr. ῥέω; ai. *srávati* ‘es fließt’

*#sn: zu uridg. *sneig^{wh}-: gr. νείφει; ahd. *snīwit* ‘es schneit’

*#su: uridg. *suekurós > gr. ἑκυρός; ai. *suaśurás* ‘Schwiegervater’

*VsV: uridg. *ǵenb₁esos > hom. γένεος; ai. *janasas* G.Sg. ‘Geburt/Erzeugung’

*Vsr: *k^bésras > ion-att. χεῖρας, lesb. χέρρας A.Pl. ‘Hand’

*Vsn: *selásnā > dor. σελάνᾱ, lesb. σελάννᾱ ‘Mond’ (Anlaut unklar)

*Vs_u: *nasuós > lak. νᾱFός, lesb. ναῦος ‘Tempel, Gotteshaus’

$s > h \setminus (R, \text{u})_{-}V$ (nach unbetonter Silbe)

*a_usós > hom. ἠώς, dor. ᾠFός, lesb. αῶως ‘Morgenröte’

*korsá > att. κουρά ‘das Abschneiden’

$s > h \setminus N_{-}V$

*k^bánsas > att. χῆνας, boiot. χᾱνας A.Pl. ‘Gans’

18 *s neben *ǵ Nach Rix §89g ist *s neben *ǵ über *h assimiliert worden. Die Beispiele belegen jedoch nur eine Assimilation bei *ǵ_i und der Weg über *h scheint nur aus systematischen Gründen gewählt zu sein.

*ǵ_i wird intervokalisch zu *ǵ_i und anlautend möglicherweise zu *ǵ_i. Für den Anlaut ist gr. ὑμῆν ‘dünne Haut, Membran’ das einzige Beispiel, das mit ai. *syúman* ‘Band, Zügel’ zusammengestellt wird. Da aber sowohl *s als auch *i auch alleine im Anlaut zu h werden und anlautendes *u ohnehin immer aspiriert ist (s. 25), ist die Zwischenstufe *ǵ_i zwischen *ǵ_i- und *h- hypothetisch.

*ǵ_is kommt intervokalisch im L.Pl. *-o_isi vor, der myk. <-o-i> geschrieben wird. Die Lautung ist unklar – vielleicht *-o_iǵi oder *-o_ihi – da später das *s analogisch restituiert wurde.

$s_{\text{ǵ}} > \text{ǵ} \setminus V_{-}V$

*gelós-ǵios > gr. γελοῖος ‘lustig, lächerlich’

$s_{\text{ǵ}} > \text{ǵ} \setminus \#_{-}V$

s.o.

? $s_{\text{ǵ}} > \text{ǵ} \setminus V_{-}V$

s.o.

19 OSTHOFF Die klassische Formulierung von OSTHOFFS Gesetz ist Kürzung von Langvokalen vor Resonant oder Halbvokal und Konsonant (Rix §58, 64). OSTHOFFS Gesetz ist jedoch umstritten und es gibt mindestens ein sicheres Gegenbeispiel, das vermutlich auf eine andere Syllabifizierung deutet. Dann wäre die Formulierung aber einfach die, dass Langvokal vor tautosyllabischem Resonanten oder Halbvokal gekürzt wird (im Auslaut ist die antevokalisches Sandhivariante verallgemeinert, so dass Langvokale erhalten bleiben). Für eine detaillierte Untersuchung s. Simkin (2004).

In att. G.Sg. $\mu\eta\nu\acute{o}\varsigma < *m\bar{e}ns\acute{o}s$ ist der Langvokal geblieben (Ersatzdehnung hätte geschlossenes $\bar{e} < \epsilon_1$ ergeben). Die beste Lösung ist die s-Verhauchung (17) vor OSTHOFF zu datieren und eine Syllabifizierung $*m\bar{e}.nb\acute{o}s$ anzunehmen, bei der das Gesetz nicht gilt. In welchen Fällen genau eine andere Syllabifizierung angenommen werden sollte, werde ich hier nicht detailliert behandeln. Simkin nimmt neben $*nb$ auch noch $*nm$ und $*\underline{ni}$ an, aber ich werde mich vorerst auf $*nb$ beschränken.

Das Lautgesetz muss vor 20 datiert werden wie z.B. hom. $\xi\sigma\tau\alpha\nu < *est\acute{a}nt < *e-steh_2-nt$ zeigt.

$\bar{V} > V \setminus _ (R, \bar{V})$. (außer im Auslaut)

uridg. $*-\bar{o}i\acute{s}$ I.Pl. o-St. $>$ gr. $-\omicron i\varsigma$; ai. $-\bar{a}i\acute{s}$

gr. $-\theta\epsilon\nu\tau-$, Stamm des Partizips Aor.Pass. auf $-\theta\eta-$.

uridg. $*e-steh_2-nt > *est\acute{a}nt >$ hom. $\xi\sigma\tau\alpha\nu$ 'sie stellten sich'

20 Konsonanten im Auslaut Wortauslautende Okklusive gehen verloren (Rix §100).

Dies muss nach der Vokalisierung der silbischen Liquiden passiert sein, da $\acute{\upsilon}\pi\acute{o}\delta\rho\alpha < *upodrak < *upodrk̂$ den auslautenden Konsonanten voraussetzt. Außerdem nach OSTHOFF (19) wie dort gezeigt.

{Okklusiv} $>$ $\emptyset \setminus _ \#$

uridg. $*-t$: uridg. $*e-b^{heret} >$ gr. $\xi\phi\epsilon\rho\epsilon$; ai. $abharat$ 'er trug'

uridg. $*-k̂$: $*upo-drk̂ > *upodrak >$ $\acute{\upsilon}\pi\acute{o}\delta\rho\alpha$ 'von unten guckend'

Auslautendes $*m$ erscheint als n (Rix §77).

Wann dies passierte, ist vollkommen unklar. Es wird hier nur zusammen mit dem obigen Gesetz gruppiert, da beide den Auslaut betreffen.

$m > n \setminus _ \#$

uridg. $*-om$ A.Sg. o-St: gr. $-\omicron\nu$, ai. $-am$, lat. $-om > -um$

21 Palatalisierung Die Anfänge der Palatalisierungen liegen wohl in urgr. Zeit und die genaue Entwicklung in den Dialekten ist kompliziert. Zu diesem Thema siehe Allen 1958, Bartoněk 2003, 140ff. und Rix §102, 103.

Linear B hat zwei Reihen für die Schreibung der späteren Zischlaute, deren Konsonanten mit <z> und <s> transliteriert werden. Welche Laute mit diesen Zeichen bezeichnet wurden, ist unklar. Oft wird <s> als einfacher Sibilant und <z> als dentale Affrikate verstanden, jedoch ist es sehr gut möglich, dass es sich tatsächlich um einen Unterschied der Palatalität handelt und beide Zeichenreihen sowohl Sibilanten als auch Affrikaten bezeichnen können. Die Z-Reihe würde dann einen palataleren Laut als die S-Reihe bezeichnen (<z> also [tʃ] und [dʒ], <s> [s], [z] und [ts]). Mit dieser Annahme lassen sich die mykenischen Schreibungen und die Lautentwicklung fast zufriedenstellend erklären. Noch besser ist vielleicht die Annahme, dass es sich bei der Z-Reihe um eine Art präpalatalen Plosiv und bei der S-Reihe um einen Sibilanten oder eine Affrikate handelt (<z> also [č], [j], <s> [s], [z], [ts], [tʃ]).

Dentale Tenuis (aspirata) + **ǵ* im Anlaut ergibt in allen Dialekten *s*, das urgr. sein könnte oder erst mit 45 im Silbenanlaut zu *s* wird. Im Inlaut wird hier **ts* angesetzt, das mit altem **ts* zusammenfällt und sich dann dialektal unterschiedlich entwickelt (s. 45). In Linear B wird für diesen Laut die S-Reihe benutzt, weswegen eine palatale Qualität sehr unwahrscheinlich ist.

(*t, tʰ*)*ǵ* > *s* \ # _

zu uridg. **tǵegw-*: gr. σέβονται 'sie sind erfürchtig'; ai. *tyajante* 'sie verlassen'

(*t, tʰ*)*ǵ* > *ts* \ _

**totǵios*: ion-att. τόσος, lesb. τόσος 'so groß'; cf. boiot. ὀπόττος, kret. ὀπόττος 'so groß wie'

**pantǵa* > **pantsa* > **pansa* (s. 45) > ark. πάνσα, ion-att. πᾶσα, lesb. παῖσα 'jede'

uridg. **médǵios* > ion-att. μέσος, lesb. μέσος, boiot. μέττος 'in der Mitte'

Dieselbe Lautkombination an morphologisch transparenter Stelle ergibt einen anderen Laut, der in alphabetischer Zeit mit dem Resultat der Palatalisierung der Velare zusammenfällt, in Linear B aber mit der S-Reihe geschrieben wird. Vielleicht wurde an dieser Stelle das **ǵ* restituiert und die Gruppe **tsǵ* nachmyk. weiterpalatalisiert, so dass sie mit Resultat der palatalisierten Velare zusammenfiel. Möglicherweise handelt es sich hier aber auch um denselben Laut, den ich unten für die Substratwörter ansetze, da Schreibung und Entwicklung mit diesem völlig übereinstimmen.

Stimmhafte Dentale und Velare fallen vor **ǵ* in einem Laut zusammen, der in Linear B mit der Z-Reihe und im Alphabetgriechischen mit <ζ> geschrieben wird. Der Laut dürfte im Ugr. entweder eine stimmhafte Affrikate oder ein (vermutlich geminiertes) stimmhafter präpalataler Plosiv gewesen sein (dafür schreibe ich hier abstrahierend <ǵǵ>). Dass hier Dentale und Velare zusammenfallen ist eigenartig. Später wird der Laut zu **dz* (s. 30).

$(d, g)\dot{i} > \check{j}$

uridg. $*d\check{i}éus > \text{gr. } ζεύς; \text{ ai. } dyáus$ ‘Himmel(sgott)’

zu ion-att. ἀρπαγή ‘Raub’: ἀρπάζω ‘ich raube’

zu uridg. $*g^{w\check{i}eh_3-: \text{ hom. } ζωή$ ‘Leben’

uridg. $*jugóm: \text{ gr. } ζυγόν; \text{ ai. } yugám$ ‘Joch’ (s. 4)

Velare Tenuis (aspirata) + $*\dot{i}$ scheint eine spätere Palatalisierung als die der Dentale zu sein. Das Ergebnis wird in Linear B ebenfalls mit der Z-Reihe geschrieben, was eine stimmlose Version des obigen Lautes nahelegt, den ich mit $\langle\check{c}\check{c}\rangle$ bezeichne. Später wird dieser Laut (vermutlich über so etwas wie $*t\check{s}$) zu ss bzw. tt (s. 54)

$(k, k^b)\dot{i} > \check{c}\check{c}$

$*k\check{i}-ámeron > \text{ att. } τήμερον, \text{ ion. } σήμερον, \text{ dor. } σάμερον$ ‘heute’

$*p^bulák\check{i}ō > \text{ att. } φύλαττω, \text{ ion. dor. } φυλάσσω$ ‘ich behüte’ zu φύλαξ ‘Wächter’

Ein anderer Laut, der in Substratwörtern vorkommt, fällt im Alphabetgriechischen völlig mit diesem $*\check{c}\check{c}$ zusammen, wird aber im Linear B mit der S-Reihe geschrieben. Da zwischen $*ts$ und dem zuerst vorgeschlagenen $*t\check{s}$ (statt $*\check{c}\check{c}$) als Ergebnis der Velar-palatalisation jedoch wenig Spielraum für eine weitere Affrikate bleibt, die zwar später mit $*t\check{s}$ zusammenfiel, aber dental genug wäre, um in Linear B mit der S-Reihe geschrieben zu werden, ist die Bestimmung der Laute der Z-Reihe als irgendwie palatale Plosive vielleicht die beste Lösung. Dann könnte man den Substratlaut als $*t\check{s}$ ansetzen und eine spätere Entwicklung $*\check{j}\check{j}$, $*\check{c}\check{c} > *d\check{z}$, $*t\check{s}$ annehmen. Heteromorphemisches $*t\check{i}$ könnte sich ebenfalls zu $*t\check{s}$ entwickelt und somit die Möglichkeit der Einordnung des Substratlautes in das Lautsystem überhaupt erst ermöglicht haben.

Labial + $*\dot{i}$ ergibt pt in allen Dialekten, muss jedoch eine Vorstufe urgr. $*p\check{c}$ (ohne eine genaue phonetische Interpretation nahelegen zu wollen) gehabt haben, da das Myk. ein eigenes Zeichen für diesen Laut kennt, der erst später auch für ererbtes pt verwendet wird (s. 28).

$\{\text{Labial}\}\dot{i} > p\check{c}$

$*klep\check{i}ō > \text{ gr. } κλέπτω$ (cf. lat. *clepō*) ‘ich stehle’

$*t^bap^h\check{i}ō > \text{ gr. } θάπτω$ ‘ich begrabe’ (cf. τάφος ‘Grab’)

Die Resonanten $*l$, $*r$, $*n$ ($*m$ in dieser Position schon mit $*n$ zusammengefallen) werden vor $*\dot{i}$ zu palatalisierten Geminaten. Ob hier Anlaut oder Silbengewicht eine Rolle spielen, ist noch zu überprüfen. Später werden die vorangehenden Vokale diphthongiert bzw. gelangt oder die Geminaten entpalatalisiert (s. 39).

$(l, r, n)\dot{i} > (\acute{l}\acute{l}, \acute{r}\acute{r}, \acute{n}\acute{n})$

$*kat^h\check{a}riō > \text{ ion-att. } καθαίρω$ ‘ich reinige’

**g^wamjō* > ion-att. βαίνω ‘ich komme’
 **áljos* > kypr. αἴλος sonst ἄλλος ‘ein anderer’
 **p^{bt}terjō* > ion-att. φθείρω, lesb. φθέρρω ‘ich zerstöre’
 **krinjō* > ion-att. κρίνω, lesb. κρίννω ‘ich entscheide’

22 **dl* > **ll* Rix §104. Datierung unklar.

d > *l* \ *_l*

uridg. **sed-leh₂*: lak. ἔλλᾱ (cf. got. *sit-ls* mit *-*los* im Suffix)

23 Metathese von **t* neben Okklusiven Nach Rix (§106b) werden **tk* und **tp* metathetisiert. Für **tp* > **pt* gibt er als Beispiel **k^wid-pe* > τίπτε, das aber wohl besser als **k^wid-k^we* > **k^wik^wte* > τίπτε zu erklären ist (Lillo 1992). Will man den in einigen Wörtern rätselhaften Anlaut *pt-* neben *p-* als Sandhi aus *-*t##p-* erklären, so wäre auch eine Metathese neben Labialen anzunehmen. So kann man die allgemeine Regel aufstellen, dass *t* + Tenuis metathetisiert wird (keine Beispiele für Media und Aspirata).
 Datierung unklar.

t(k, k^w, p) > (*k, k^w, p*)*t*

**tk*: **ti-tk-ō* zur Wurzel uridg. **teḱ-* ‘gebären’ > τίκτω ‘gebiert’

**tk^w*: **k^wid-k^we* > **k^wik^wte* > τίπτε ‘warum denn?’

? **tp*: *-*t##pólis* > πτόλις

24 Nasal vor Liquida Zwischen Nasal und Liquida wird die homorgane Media eingeschoben. Im Anlaut ist solch eine Lautverbindung unmöglich und der Nasal wird stattdessen schlicht zur Media (Rix §78).

Die Datierung hier ist noch unsicher. Sogar die Einordnung als urgr. ist nur vorläufig.

NR > *DR* \ #*_*

uridg. **mṛtós* > aiol. βροτός ‘Sterblicher’; ai. *mṛtás* ‘tod’

**m_{h3}skoh₂* > βλώσκω ‘ich komme’ zu μολεῖν ‘kommen’

NR > *NDR*

uridg. **h₂nrós* > **anrós* > ἀνδρός G.Sg. zu ἀνήρ ‘Mann’

hom. μέ-μβλεται ‘er ist besorgt’ zu μέλω ‘ich Sorge mich’

25 **u-* > *bu-* Im Anlaut ist **u* immer behaucht. Da alle (nicht psilotischen, s. 56) Dialekte, dieses Phänomen zeigen, setze ich es als urgr. an. Dass es nach GRASSMANN (s. 55) zu datieren ist, wie Lejeune (§320) wegen ὑφαίνω (cf. ai. *ubhnāti*) behauptet, ist nicht zwingend, da bei GRASSMANN der erste Laut deaspiert wird und ein bloßes **u* im Anlaut im Gr. nicht vorkommt.

$u > hu \setminus \# _$

uridg. **upo* > gr. ὑπό; ai. *úpa* ‘unter’

Das Lautsystem des Urgriechischen

Das urgr. Lautsystem stellt sich nach den obigen Lautgesetzen nun wie folgt dar:

Vokale: **a*, **ā*, **e*, **ē*, **o*, **ō*, **i*, **ī*, **u*, **ū*, **ə*

Resonanten und Halbvokale: **r*, **rr*, **l*, **ll*, **m*, **n*, **nn*; **ɰ*, **ɲ*

Verschlusslaute:

Labiovelare **k^w* **g^w* **k^{wb}*

Velare **k* **g* **k^b*

Dentale **t* **d* **t^b*

Labiale **p* **b* **p^b*

Palatale: **ɟ*, **č*, **pč*

Frikative und Affrikaten: **h*, **s*, **z*, **ts*, **tš*

5.4. Urgriechisch bis mykenische Zeit

In diesem Abschnitt wird die Lautentwicklung bis zur Zeit, in der die mykenische Überlieferung endet, modelliert. Am Anfang dieser Zeit steht die erste dialektale Aufspaltung. Da außer dem recht einheitlichen myk. Dialekt keine weiteren aus dieser Zeit überliefert sind, wird der Lautstand des Myk. – sofern es keinen Grund gibt, etwas anderes anzunehmen – als repräsentativ für das gesamte gr. Sprachgebiet gesehen.

Die dialektale Gliederung ist im wesentlichen aus Bartoněk 1991 Tab. B genommen und sieht in Scheme wie folgt aus (es wird jedoch nicht auf alle Sprachen auch eingegangen):

```
(urgr1 (urgr (nwgr (dor (nwdor phok lokr aitōl)
              (sardor kor meg oarg)
              (sdor el lak inseldor warg kret))
        (aiōl boiot (thess wthess othess) lesb))
      (sogr (ach myk ark kypr pamph)
            (ion-att (ostion hom) inselion euboi att))))
```

26 **ti* > *si* Dieser Lautwandel wird als grundlegende Unterscheidung zwischen Südost- und Nordwestgriechisch gesehen (Rix §101), ist aber weniger Lautgesetz als einem lieb wäre. Auch nwgr. Dialekte haben *si* in *ti*-Abstrakta (z.B. βάσις vs. ai. *gátis* ‘Gang’) und das Myk. geht in z.B. *ko-ri-si-o* weiter als andere Dialekte, die alle Κορίνθιοι zeigen (Bartoněk 2003, S. 144). In dieser Darstellung wird der Lautwandel

wegen z.B. ἔστι, κτίσις nach **s* und **k* ausgesetzt. Eine genauere Lautumgebung – falls man sie überhaupt angeben kann – ist unsicher. Vermutlich ist der Wandel schlicht nicht lautgesetzlich sondern als eine dialektal unterschiedliche Verallgemeinerung von Sandhivarianten zu sehen.

$t > s \setminus _i$ außer nach *s, k* (nwgr.)

uridg. **b^héronti* > ion-att. φέρουσι, dor. φέροντι ‘sie tragen’

uridg. **b^heh₂ti* > ion-att. φησί, dor. φᾶτί ‘er sagt’

s.o.

27 **ə > *a/*o* Der Sprossvokal, der neben/aus Liquida bzw. Nasalis sonans entstanden ist, wird dialektal zu *a* bzw. *o*. Da die genaue dialektale Verteilung schwierig ist (im Ark.-kypr. und Myk. finden sich beide Vokalisationen), nehme ich hier als ungefähre Annäherung für aiolisch *o*, ansonsten *a* an.

$ə > o$ (aiol.)

$ə > a$

lesb. boiot. στροτός vs. att. στρατός ‘Truppe, Heer’

thess. πετρο- myk. *qe-to-ro* vs. att. τετρα- ‘vier’

myk. *a-mo-ta* vs. att. ἄρματα ‘Wagen’

lesb. ark. δέκοτον myk. *de-ko-to* vs. att. δέκατον ‘das zehnte’

28 **pč > pt* Nach Bartoněk (2003, S. 106) waren die Laute im Myk. schon zusammengefallen, da das PTE-Zeichen oft mit <pe-te> alterniert. Dies kann man jedoch auch so deuten, dass die Laute eben noch nicht ganz zusammengefallen waren (<pte> also noch für etwas wie **pč* stand) und erst im Laufe der myk. Überlieferung eine Schreibung <pe-te> für ursprüngliches **pč* bzw. <pte> für ursprüngliches *pte* möglich wurde. Wie auch immer die genauere Datierung sein mag, lange vor myk. Zeit kann sie nicht sein, da die Palatalisierungen eher spät in urgr. Zeit sein dürften und ein allzu plötzlicher Wandel **pč > pt* unwahrscheinlich scheint.

$pč > pt$

s. Beispiele bei 21

29 **ĵ > *b* Anlautendes und intervokalisches **ĵ* wird im Myk. teilweise mit der J-Reihe, teilweise (wenn vorhanden) mit einem H-Zeichen, und teilweise gar nicht geschrieben. Daraus ist zu schließen, dass die Verhauchung im Laufe der mykenischen Überlieferung passierte.

$ĵ > b \setminus (\#, V)_V$

myk. *jo* und *o* = att. ὥς ‘wie’

5.5. Nachmykenische/alphabetische Zeit

In dieser Zeit bilden sich die meisten Unterschiede der Dialekte heraus. Durch die Fülle an Dialekten und der dazu im Vergleich eher spärlichen Belegsituation sind die genauen Verhältnisse in vielen Fällen unklar und die hier aufgestellten Regeln entsprechend mit mehr oder weniger großer Unsicherheit behaftet. In fast jedem Fall müsste man sich eingehend mit den Dialektinschriften auseinandersetzen, um das Phänomen so gut es geht beschreiben zu können. In vielen Fällen sind die Lautwandel zwar recht klar, aber wie sie dialektal gestreut sind und wann sie zu datieren sind, ist oft sehr unsicher.

Ein Hauptproblem bei der Überlieferung ist, dass viele Dialekte erst in späterer Zeit überliefert sind, in der man schon mit Beeinflussung durch die ion-att. Koiné rechnen muss. Daher weiß man oft nicht, ob ein Merkmal in einem Dialekt ererbt oder entlehnt ist.

Letztendlich ist es nicht mein Ziel, die exakte Lautung jedes belegten Dialektes aus dem Ugr. herzuleiten, da diese Aufgabe wegen des zeitlich und räumlich gesäten Variantenreichtums in dieser Arbeit nicht zu schaffen ist. Stattdessen werde ich mich auf den Lautstand der etwas größeren Dialektgebiete zur Zeit des Einsetzens ihrer Überlieferung konzentrieren (auch wenn sich dieser Zeitpunkt von Dialekt zu Dialekt recht stark unterscheiden kann).

Die eingeschobenen ungefähren Jahreszahlen habe ich anhand einiger Termini post quos für die Dialektaufspaltung, die ich aus den von den Lautwandeln betroffenen Dialekten geschlossen habe, aus dem Dialektstammbaum von Bartoněk (1991) eingefügt. Einige Zeiträume scheinen merkwürdig ereignislos, andere wiederum sehr bewegt. Auch hier ist also wohl noch keine Wahrheit gefunden und an Datierung sowohl der dialektalen Aufspaltung als auch der Lautgesetze sollte noch gearbeitet werden.

30 *čč̣, *jŷ *čč̣ fällt mit *tš zusammen, die im Myk. (unter der Annahme, dass meine Deutung der Schreibungen korrekt ist) noch getrennt sind. Da es außerhalb des Myk. keinen Hinweis auf ursprünglich zwei verschiedene Laute gibt, setze ich dieses und das nächste Lautgesetz so früh wie möglich an.

čč̣ > tš

*jŷ wird (vielleicht über *dẓ̌) zur dentalen Affrikata *dz, die in den Regeln, da wir bei s die Stimmhaftigkeit ignorieren, mit ds bezeichnet ist. Die genaue spätere Lautung ist umstritten. Geschrieben wird der Laut mit <ζ>, das auf jeden Fall [zd] bezeichnen konnte, was Wörter zeigen, bei denen *zd ursprünglich ist (z.B. ὄζος = got. *asts* 'Ast'). Eine Metathese *zd > *dz ist unwahrscheinlich und einer Erhaltung von *dz < *(d, g)i widerspricht ἔρδω 'machen' < *u_{er}zdō < *u_{er}dzō < *u_{er}jŷō < *u_{er}giō. *dz wird also nur als Zwischenstufe zwischen *jŷ und *zd angenommen.

$\check{y} > ds$

31 el. $\bar{e} > \bar{a}$

Im Elischen ist \bar{e} mit <ε> aber auch oft mit <α> wiedergegeben, was eine sehr offene Aussprache nahelegt. Für Bartoněk (1966, 89ff.) ist dieser Lautwandel sehr alt und er setzt ihn vor dem 1. Jt. an.

$\bar{e} > \bar{a}$ (el.)

zu uridg. $*d\check{i}em$ A.Sg. ‘Himmel(sgott)’: el. Ζᾶνες N.Pl. ‘Zeusstatuen’ vs. hom. Ζῆνα A.Sg. ‘Zeus’

el. μᾶ neben μή ‘nicht’

el. Φράτρᾱ vs. lak. ῥήτρᾱ ‘Verabredung’

32 Labiovelare Die genaue Entwicklung der Labiovelare ist in einigen Dialekten etwas unklar. Gemeingriechisch ist der Wandel {Labiovelar} > {Labial}. Darüber hinaus haben die meisten Dialekte einen Wandel {Labiovelar} > {Dental} in bestimmten Umgebungen.

Das Aiolische hat nur ausnahms- und unerklärterweise dentalen Reflex. Das Ark. und Kypr. haben von den restlichen Dialekten etwas andere bzw. weitergehende Entwicklungen, auf die ich allerdings in dieser vereinfachten Darstellung nicht detailliert eingehen werde. Rix §96-99.

Da sich alle aiol. Dialekte gleich verhalten, ist der Terminus ante quem für die Entwicklung zu Dentalen wohl vor der Aufspaltung des Uraiol. zu sehen, den Bartoněk (1991) auf etwa 1100 datiert. Da sich das Ark. und Kypr. vor e anders verhalten, dürfte der Wandel nach Aufspaltung dieser beiden Dialekte passiert sein, den Bartoněk (1991) auf etwa 1300 datiert. Die Entwicklung zu Labialen muss etwas später sein, erfasst aber den gesamten Sprachraum, und zwar vor Beginn der alphabetischen Überlieferung. Damit ist hierfür der Terminus ante quem in etwa das 9. Jh.

$k^w > t \setminus _i$ (nicht aiol.)

uridg. $*k^wis > \text{gr. τίς}$ (auch aiol.); lat. *quis* ‘wer?’

ion-att. τίσις ‘Vergeltung’ neben ποινή ‘Buße’

{Labiovelar} > {Dental} $\setminus _e$ (nicht aiol. kypr.)

uridg. $*-k^we > \text{gr. -τε}$ (auch aiol.); lat. *-que* ‘und’

uridg. $*pénk^we > \text{aiol. πέμπε}$, sonst πέντε ‘fünf’

uridg. $*g^w elb^bús > \text{gr. δελφύς}$ ‘Mutterleib’; Δελφοί neben boiot. Βελφοί

zu uridg. $*g^w ben-/g^w bon-$: gr. θείνω ‘schlagen’ neben φόνος ‘Mord’

{Labiovelar} > {Labial}

uridg. $*pénk^wtos > \text{gr. πέμπτος}$ ‘der fünfte’

**g^wh₂iō > *g^wám₂iō > gr. βαίνω; lat. veniō ‘ich komme’*
s.o. φόνος

Nach dem Wandel assimilieren sich die Nasale wieder an den folgenden Laut (z.B. in πέμπτος). Dies ist wohl auch als phonologische Regel anzusetzen.

33 **b, *g > m, ŋ*

**b* und **g* assimilieren sich an ein folgendes *n* und werden zu Nasalen (Rix §105), wobei zu beachten ist, dass der velare Nasal von der velaren Media graphisch nicht unterschieden ist (beide <γ>).

Dies muss, wenn man keinen labiovelaren Nasal annehmen will, nach der Beseitigung der Labiovelare passiert sein.

b, g > m, ŋ \ _n
**tieg^wnós > *sebnós > σεμνός ‘ehrfürchtig’*

ca. 1000

34 **{Dental}t₂* **t₂* verhält sich im Anlaut wie **t₂*, sonst wie **k₂* (Rix §104). Eine Hesychglosse erweist unverändertes *t₂e* (<τρε>) fürs Kretische. Dass **t₂* nicht komplett mit einer der beiden Lautgruppen zusammenfällt, ist etwas überraschend.

Nach Rix bleibt **t^b₂* zunächst erhalten. Da Beispiele fehlen, könnte man aber auch (zumindest im Inlaut) als Ergebnis *st^b* annehmen, wodurch man die Endung der 2.Pl. -σθε mit ai. -*dhvam* verbinden kann.

Eine genaue Datierung ist unklar.

t₂ > s \ #_ (nicht kret.?)
uridg. **t₂e > ion-att. σε; cf. ai. tvām ‘dich’*
**t₂e₂sō > gr. σείω ‘erschüttern’*

t₂ > tš
hom. ἐσσεῖοντο zu σείω, cf. ai. *atviṣanta* ‘sie gerieten in Erregung’.
**k^wet₂res > ion. τέσσαρες, att. τέτταρες, boiot. πέτταρες ‘vier’*

? *t^b₂ > st^b*
s.o.

35 **dz > zd*

Wie schon bei 30 erklärt, wird **dz* zu *zd* <ζ>.

Dies muss wegen ἔρδω vor 37 passieren.

ds > sd

36 $*(N, \{Dental\})sC$ Nasal und Dental schwinden vor $*sC$. Dies muss vor 37, wie att. ἴσος < $*\text{u}its\text{u}os$ ‘vielleicht’ und δεσπότης < $*dems-p\acute{o}t\acute{a}s$ ‘Hausherr’ zeigen, und nach 35 passieren, wie σύζυγος < $*sun-zdugos$ < $*sun-dzugos$ ‘zusammengespannt’ zeigt. Rix §78.

$(N, \{Dental\}) > \emptyset \setminus _sC$

$*kent-t\acute{o}s > *kenst\acute{o}s > \text{gr. } \kappa\epsilon\sigma\tau\acute{o}s$ ‘bestickt’

s.o.

37 $*CsC$ Zwischen Konsonanten wird $*s$ in der Regel zu $*h$ verhaucht und schwindet dann ggf. mit Hauchumsprung (s. 42) (Rix §87c, Lejeune §132, 133). Die genaue Entwicklung ist nicht völlig klar, aber zwischen Resonant oder Halbvokal und stimmlosem Okklusiv scheint $*s$ erhalten zu sein. Bestimmte Lautgruppen sind dann jedoch dialektal teilweise vereinfacht worden. Eine Tenuis vor dem $*s$ wurde aspiriert, woran sich eine folgende Tenuis assimilierte. Eine folgende Media hatte Assimilation der Aspirata an die Media als Folge.

Dies muss – wie schon gesagt – nach 35 und 36 passieren. Im Myk. ist die Lautverbindung noch intakt.

$s > h \setminus C_C$ (außer in $(R, \text{V})_ (T, T^b)$)

uridg. $*p\acute{e}rsneh_2$; ion-att. πτέρνη; got. *fáirzna* ‘Ferse’

$*orsm\acute{a} > \text{ion-att. } \acute{o}p\mu\eta$ ‘Ansturm, sich in Bewegung setzen’

aber:

$*\text{u}oid-t^ba > *\text{u}oist^ba > \text{att. } \omicron\iota\sigma\theta\alpha$

$*par-st\acute{a}des > \text{παστάδες neben παρτάδες}$ ‘Vorhallen’

$*t^burst^ben > \text{θύρθεν neben θύσθεν}$ ‘draußen (vor der Tür)’

$T^b > T^b$

myk. *ai-ka-sa-ma* /*aĩksmá*/ = hom. αἰχμή ‘Speer’

$T > T^b \setminus T^b_$

$*eks-tr\acute{o}s > \text{gr. } \acute{\epsilon}\chi\theta\rho\acute{o}s$ ‘Fremder, Feind’

$T^b > D \setminus _D$

zu uridg. $*pesd-/psd-$: gr. βδέω ‘furzen’

38 $*\text{u}i > *\ddot{u}\ddot{u}$ $*\text{u}i$ wird zu einer palatalen Geminata. Damit gliedert sie sich in die Reihe der palatalen $*\acute{l}\acute{l}$, $*r\acute{r}$, $*n\acute{n}$ ein, mit denen sie sich weitgehend parallel entwickelt (s. 39, 40). Rix §73.

$\text{u}i > \ddot{u}\ddot{u}$

$*kau-i\acute{o} > \text{hom. } \kappa\acute{\alpha}\iota\omega, \text{ att. } \kappa\acute{\alpha}\omega$ ‘ich entzünde’; cf. καῦμα ‘brennende Hitze’

**au̯i-etós* > hom. αἰετός, att. ἄετός ‘Adler’; cf. lat. *avis* ‘Vogel’
díu̯ios (myk. *di-wi-jo*) > ion-att. δῖος ‘göttlich’

39 Palatale Geminaten Nach *a* oder *o* spalten sich **rr̥*, **nn̥* (Rix §70a), **ww̥* (Rix §73) in *ɹ* und ihre ungeminierte unpalatale Entsprechung (*iɹ* entwickelt sich dialektal später zu *ji* weiter, s. 50). Im Kypr. ist das auch bei **ll̥* der Fall (zumindest nach *a*). Ansonsten wird **ll̥* entpalatalisiert und bleibt Geminat (Rix §70c). Im Thess. und Lesb. werden **rr̥*, **nn̥* in allen anderen Fällen entpalatalisiert (in den anderen Dialekten Ersatzdehnung, s. 40) (Rix §70b).

rr̥, *nn̥*, *ww̥* > *ɹr*, *ɹn*, *ɹu* \ (*a*, *o*)_

**kat̥hariō* > καθαίρω ‘ich reinige’

**gʷniō* > **gʷaniō* > βαίνω ‘ich komme’

**au̯ietós* > αἰβετός (Hesychglosse), hom. αἰετός ‘Adler’

**kom-iós* (cf. lat. *cum* ‘mit’) > κοινός ‘gemein(sam)’

ll̥ > *ɹl* \ (*a*, *o*)_ (kypr.)

ll̥ > *ll*

uridg. **ál̥ios* > kypr. αἴλος, sonst ἄλλος; lat. *alius* ‘ein anderer’

rr̥, *nn̥* > *rr*, *nn* (thess. lesb.)

**p̥t̥heriō* > lesb. φθέρρω, ion-att. φθείρω ‘ich zerstöre’

**oiktir̥iō* > lesb. οἰκτίρρω, ion-att. οἰκτίρω ‘ich bedauere’

**kten̥iō* > lesb. κτέννω, ion-att. κτείνω ‘ich töte’

**krin̥iō* > lesb. κρίννω, ion-att. κρίνω ‘ich entscheide, wähle’

40 Erste Ersatzdehnung Die erste Ersatzdehnung dehnt grob gesprochen Vokale vor palatalen Geminaten (Rix §70b, 73), **b* + Resonant (Rix §88d) und **ln* (Rix §77) außer im Thess. und Lesb., wo es zur Geminatbildung kommt (s. 43). Gedeckte *e*, *o* fallen dabei nur in einigen Dialekten mit alten *ē*, *ō* zusammen. Im Ion-att., Nwdor. und Sardor. entstehen dabei neue geschlossene Langvokale **ē̄*, **ō̄* (Bartoněk 1966).

Datiert werden muss der Wandel vor der Hebung von **ā* (47) im Ion-att.

(*e*, *i*, *u*)*ṘṘ* > (*ē̄*, *ī̄*, *ū̄*)*R* (mit *Ṙ* = *ɹ*, *ɳ*, *ʷ*)

s.o. 39

**plun̥iō* > ion-att. πλύνω ‘ich wasche’

díu̯ios (myk. *di-wi-jo*) > ion-att. δῖος ‘göttlich’

zu εὐρύς: **eur̥éu-ia* > hom. att. εὐρεῖα (wohl /*ē̄*/, nicht /*eṛ̥*/), lesb. εὐρηα ‘breit’

VRh, *VhR* > *VR* \ _*V* (mit *R* = Resonant oder *u*; nicht thess. lesb.)

s. die Beispiele bei 17

$Vln > \bar{V}l \setminus _n$ (nicht thess. lesb.)

$Vln > Vll$

$*g^{w}oln\acute{a} >$ ion-att. βουλή, ark. βωλᾶ́, lesb. βόλλᾱ ‘Wille, Rat’

41 $*\underline{u}s > *\underline{u}b$ Der Unterschied, den die Betonung bei $*\underline{u}s$ offenbar ausmacht, verdient eine gesonderte Betrachtung. Die Rekonstrukte uridg. $*h_2eus\acute{o}s$ ‘Morgenröte’ und $*h_1\acute{e}usoh_2$ ‘ich senge’ haben trotz ihrer fast gleichen Lautung die sehr unterschiedlichen Reflexe att. ἔως, hom. ἦώς, dor. ἄFώς, lesb. αὔως und att. εὔω. die Morgenröte hat außer in att. ἔως keine Behauchung und ist für Sommer (1905, S. 12) von ἑσπέρᾱ ‘Abend’ übernommen (wobei das Lesb. und ursprünglich wohl auch das Hom. psilotisch sind – s. 56 – und daher keine sehr zuverlässige Aussage machen). Ansonsten ist Ersatzdehnung (außer im Lesb., wo immer geminiert wird) zu beobachten. Att. εὔω hingegen hat keine Ersatzdehnung sondern Geminatio und auch die Behauchung kann kaum von einem anderen Wort kommend erklärt werden. Die einzig plausible Lösung scheint, den Unterschied im Akzent zu sehen, was Wackernagel (1888) zuerst schon bei rs beobachtete (vgl. κόρη (att. κόρη) ‘Tempel, Teil der Stirn’ vs. κουρά ‘Schnitt’). Dass in der Stellung nach Akzent das $*s$ stabiler ist als vor Akzent, zeigt bei r das Beispiel oben. Daher liegt die Vermutung nahe, dass in $*\underline{u}s$ das $*s$ nach unbetonter Silbe schon früh verhaucht wurde (s. 17), dann der Ersatzdehnung (bzw. Geminatio) unterliegt (s. 40), nach betonter Silbe aber zunächst erhalten bleibt. Später wird erhaltenes $*\underline{u}s$ zu $\underline{u}u$ geminiert (s. 43), wie att. εὔω zeigt. Da hier offenbar aber zusätzlich (!) der Hauchumsprung (s. 42) stattfindet, ist eine vorherige Verhauchung des $*s$ anzunehmen.

Eine nette Bestätigung dieses Ergebnisses sind die Wörter ion-att. ἀκούω /akóuō/ < $*akósō$ ‘ich höre’ und hom. ἀκουή /akōē/, att. ἀκοή /akoē/ < $*akousá$ ‘Gehör’.

$s > b \setminus \underline{u}$

42 Hauchumsprung Die genauen Bedingungen für den Hauchumsprung sind etwas unklar. Grob gesagt springt ein b direkt nach der ersten Silbe in den Anlaut derselben. Wenn das b intervokalisch ist, geschieht dies nur, wenn die zweite Silbe unbetont ist (Sommer 1905, S. 10). Zwischen Resonant oder \underline{u} und Vokal scheint b nicht nur in den Anlaut der ersten Silbe zu springen, sondern auch am ursprünglichen Ort zu bleiben, anders ist att. εὔω < $*\acute{e}usō$ nicht verständlich.

Die Formulierung klingt nicht besonders überzeugend und möglicherweise sind die Regeln dialektal auch unterschiedlich.

$V_1bV_2 > bV_1V_2$ (wenn V_2 unbetont)

$*\acute{e}\text{-}serpont >$ εἶρπον ‘sie schlichen’

*iserós > ἱερός ‘heilig’

aber: *iháomai > att. ἰάομαι ‘ich heile’

$V(R, \underline{u})bV > *bV(R, \underline{u})bV$

*orsmá > ion-att. ὀρμή ‘Ansturm, in Bewegung setzen’

*éusō > att. εὖω ‘ich senge’

43 Gemination Zwischen Vokalen assimiliert sich *b neben Resonanten oder \underline{u} an diesen Laut, so dass eine Geminale entsteht. Hauptsächlich betrifft dieses Gesetz das Lesb. und Thess., da diese Dialekte nicht von der 1. Ersatzdehnung (40) betroffen waren, aber da * $\underline{u}b < *u_s$ auch in anderen Dialekten wirkt, scheint es sinnvoll, das Gesetz für alle Dialekte zu formulieren.

$(R, \underline{u})b > (RR, \underline{uu}) \setminus V_V$

s. die Beispiele bei 17

s. 41 zu *éusō > att. εὖω ‘ich senge’

44 *b > Ø *b schwindet außer im Anlaut.

$b > \emptyset$ (außer #_)

45 *ts *ts wird nach Konsonant (also auch nach Diphthong) zu s vereinfacht. Vor Konsonant ist dies schon bei 36 passiert.

Nach Vokal wird es im Boiot. und Kret. zu tt, sonst überall zunächst zu ss, das dann in allen Dialekten nach Langvokal, im Ion-att. und Ark. überall vereinfacht wird (bei Homer als Archaismus oder Aiolismus aber oft nicht). Rix §102, Sihler 196.

$ts > s \setminus C_$

*pántia > *pántsa > *pánsa > thess. kret. ark. πάνσα, lesb. παῖσα, ion-att. πᾶσα (s. 48) ‘jede’

$ts > tt$ (boiot. kret.)

$ts > ss$

boiot. ὀπόττος, kret. ὀπόττος ‘so groß wie’; ion-att. τόσος, lesb. τόσος ‘so groß’
uridg. *médbhios > ion-att. μέσος, lesb. μέσος, boiot. kret. μέττος ‘in der Mitte’

$ss > s \setminus \bar{V}_$

hom. τάπησι (ταπητ- + -σι) ‘auf den Teppichen’ vs. hom. ποοσί/ποσί (ποδ- + -σι)
‘an den Füßen’

$ss > s$ (ion-att. ark.)

s.o. μέσος &c.

46 $zd > d(d)$ In einigen Dialekten wird zd zu dd , im Anlaut zu d (Rix §102, 103).

$zd > dd$ (boiot. kret. lak. el.)

Präsentien in $-\acute{\alpha}\delta\delta\omega$, $-\acute{\iota}\delta\delta\omega$ vs. $-\acute{\alpha}\zeta\omega$, $-\acute{\iota}\zeta\omega$

$dd > d \setminus \#_$

boiot. $\Delta\epsilon\acute{\upsilon}\varsigma$ = ion-att. $\text{Ζε}\acute{\upsilon}\varsigma$; kret. $\Delta\eta\nu\alpha$ = hom. $\text{Ζ}\eta\nu\alpha$

ca. 900

47 $\bar{a} > \text{ion-att. } \bar{e}$ Das offensichtlichste Merkmal der ion-att. Dialekte. \bar{a} wird gehoben, fällt jedoch noch nicht mit ererbtem \bar{e} zusammen, da die beiden Laute im Inselion. noch mit unterschiedlichen Zeichen geschrieben werden.

Der Lautwandel ist zwischen der ersten und zweiten Ersatzdehnung anzusetzen, da durch die erste Ersatzdehnung entstandenes \bar{a} den Lautwandel mitmacht und bei der zweiten ein neues \bar{a} entsteht.

$\bar{a} > \bar{e}$ (ion-att.)

ion-att. $\mu\acute{\eta}\tau\eta\rho$, sonst $\mu\acute{\alpha}\tau\eta\rho$ 'Mutter'

In der Nikandre-Inschrift: $\langle \text{Νικανδρη} \rangle$ mit $\langle \eta \rangle$ für $/\bar{e}/$ aber $\langle \epsilon\theta\epsilon\kappa\epsilon\nu \rangle$ (klassisch $\xi\theta\eta\kappa\epsilon\nu$) mit $\langle \epsilon \rangle$ für $/\bar{e}/$ (sonst beide ion. $\langle \eta \rangle$)

ca. 800

48 Zweite Ersatzdehnung

Bei der zweiten Ersatzdehnung schwindet n in der Lautgruppe Vokal + ns , das entweder im Auslaut steht und somit nicht der ersten Ersatzdehnung (s. 40) unterlag, oder im Inlaut mit neuentstandenen $s < *ts$.

Die zweite Ersatzdehnung betrifft nicht alle Dialekte in gleicher Art. In einigen wird der Vokal gedehnt, in anderen wird er zum i-Diphthong, und in einigen unterbleibt der n-Schwund und damit die Ersatzdehnung völlig (Rix §78). Das genaue Ergebnis kann in- und auslautend unterschiedlich sein und auch vom Vokal abhängen. Auslautendes $-Vns$ konnte schon durch 36 als antekonsonantische Sandhivariante zu $-V\bar{s}$ werden und verallgemeinert werden, so dass einige Dialekte zur Zeit der zweiten Ersatzdehnung $-Vns$ schon ganz aufgegeben hatten.

Die Ergebnisse von gedehnten e , o sind im Ion-att., Nwdor. und Inseldor. die geschlossenen Langvokale \bar{e} , \bar{o} , ansonsten fallen sie mit den alten offenen zusammen.

Die genauen Regeln sind also nicht ganz einfach zu formulieren. Im wesentlichen halte ich mich an die Karte, die bei Bartoněk (1991) abgedruckt ist (ursprünglich von Hainsworth, *Cambridge Ancient History* III 1, 863).

$Vns > V_{\tilde{z}}s \setminus _V$ (lesb.)

$Vns > \bar{V}s \setminus _V$ (ion-att. boiot. el. lak. inseldor. kypr.)

*-ontia (Partizip N.Sg.f.) > *-onsa > kret. -ονσα; lesb. -οισα; ion-att. -ουσα, lak. -ωσα

$Vns > V_{\tilde{z}}s \setminus _ \#$ (lesb. el.)

$Vns > \bar{V}s \setminus _ \#$ (ion-att. boiot. lak. inseldor.)

uridg. *tóns (Artikel A.Pl.) > kret. arg. τόνς; lesb. τοίς; ion-att. τούς, lak. τώς

49 Kontraktionen Einige Kontraktionen passieren vor dem Digammaschwund, sind aber teilweise unklar und schwer zu formulieren und werden daher hier übersprungen.

50 Digamma Bei γ ist die dialektale Entwicklung äußerst schwierig. Teilweise geht es vor der Überlieferung verloren (z.B. ion-att.), teilweise ist es bis heute im Tsakonischen bewahrt. Grundsätzlich schwindet γ in vielen Dialekten zwischen Vokalen (dort besonders oft), im Anlaut (vor Vokalen und Konsonanten) und nach Konsonanten (unter 53 behandelt) (Rix §71, 72, 73). Nur bei den Dialekten, in denen seit dem Beginn der Überlieferung γ fehlt, wird hier der Schwund angesetzt. Die hier formulierten Regeln beruhen hauptsächlich auf Buck (1910) und Lejeune (1972) und ihre Vorläufigkeit und Unvollständigkeit kann nicht genug betont werden. Ehrlicherweise müsste man diesen Abschnitt zunächst überspringen, jedoch ist er zu wichtig um völlig ignoriert zu werden. Eine genaue Untersuchung wäre gerade zu diesem Thema sehr wünschenswert. Der (intervokalische) Digammaschwund hängt eng mit den Kontraktionen zusammen; eine genauere Untersuchung muss also beide Phänomene betrachten.

Manchmal wird γ im Anlaut zu h statt ganz zu schwinden, ohne dass die Bedingungen genau bekannt wären.

$\gamma > \emptyset \setminus V_V$ (ion-att. inseldor. lesb. kret. ark.)

uridg. *néγos: myk. ne-wo, kypr. νεφοσ-, sonst νέος 'jung, neu'

$\gamma > h \setminus \#_VsC$ (att.)

*γεσπέρᾱ > att. ἑσπέρᾱ 'Abend'; cf. lat. *vesper*

$\gamma > \emptyset \setminus \#_V$ (ion-att. inseldor. lesb.)

*γóikos > οἶκος, sonst φοῖκος 'Haus' (cf. lat. *vīcus* 'Dorf')

$\dot{\gamma}\gamma > \dot{\gamma}\dot{\gamma} \setminus V_V$ (ion-att. inseldor. lesb.)

*aḡγesi > kypr. lokr. αἰΓεί, hom. αἰεί 'immer (← zu allen Zeiten)'

*ἐλαγγον (entlehnt in lat. *oleum*) > myk. e-ra₃-wo, ion-att. ἔλαιον 'Olivenöl'

**eláiyā* (entlehnt in lat. *olīva*) > myk *e-ra-wa*, hom. ἐλαίη ‘Olive’

**aij-etós* > **aijetós* > hom. αἰετός; αἰβετός (Hesychglosse)

u > $\emptyset \setminus h_V$ (ion-att. inseldor. lesb.)

uridg. **sʰekurós* > ἑκυρός, ai. *svaśurás* ‘Schwiegervater’

u > *h* \ #_r (ion-att. inseldor.)

**urétrā* > att. ῥήτρᾱ, el. Φράτρᾱ ‘Verabredung’

51 #hR Vor *l*, *m*, *n* schwindet *h* im Ion-att. Wann und in welchen Dialekten dies noch geschieht, bleibt herauszufinden.

h > $\emptyset \setminus \#_ (l, m, n)$ (ion-att.)

ion-att. λαβών neben aigin. ληαβόν ‘genommen habend’

ion-att. μία < **smía* ‘eine’ zu uridg. **sem-* ‘ein’

ion-att. νείφει ‘es schneit’ zu uridg. **sneig^{wb}*- ‘schneien’

52 att. \bar{e} > \bar{a} Das wichtigste Merkmal, das das Attische vom Ionischen unterscheidet. Nach *e*, *i*, *r* wird gehobenes \bar{e} zurück zu \bar{a} gesenkt. Dieser Wandel muss zwischen 50 und 53 datiert werden, wie die Beispiele zeigen.

\bar{e} > $\bar{a} \setminus (e, \bar{e}, \bar{e}, i, i, \bar{i}, r)_$

**néyā* > att. νέᾱ, ion. νέη ‘jung, neu (f.)’

*-*íā* > att. -ίᾱ ion. -ίη in z.B. σοφίᾱ ‘Klugheit’

**kʰórā* > att. χώρᾱ ion. χώρη ‘Platz, Raum, Land(schaft)’

aber **kórūā* > att. κόρη, ostion. κόυρη ‘Mädchen’

ca. 700

53 Dritte Ersatzdehnung und Kontraktion In der Lautgruppe Resonant, *d*, *t^b* oder *s* + *u* schwindet das *u* in vielen Dialekten. In einigen Dialekten geht das *u* spurlos verloren, in anderen wird vorangehender Vokal ersatzgedehnt und in anderen bleibt die Lautkombination intakt (Rix §72, 104b; Lejeune §71, 131). Das Ergebnis von gedehnten *e*, *o* sind die geschlossenen \bar{e} , \bar{o} im Ion-att., Nwdor. und Sardor (Bartoněk 1966).

Dasselbe Ergebnis haben die Kontraktionen von *e* + *e* und *o* + *o*.

(#, *V*)(*r*, *l*, *n*, *d*, *t^b*, *s*)*u* > (#, \bar{V})(*r*, *l*, *n*, *d*, *t^b*, *s*) (inselion., ostion., inseldor., kret.)

u > $\emptyset \setminus (r, l, n, d, t^b, s)_$ (ion-att. lesb. lak.)

**kórūā* > att. κόρη, ion. κούρη, kor. κόρFᾱ, ‘Mädchen’

**kalūós* > att. καλός, ostion. καλός, boiot. καλFός, ‘schön’

**ksénuōs* (myk. *ke-se-nu-wo*) > att. ξένος, ostion. ξείνος, kor. ξένφος ‘Fremder, Gast’

**dédymen* > att. δέδιμεν, hom. δειδιμεν ‘wir fürchten’

**duís* > hom. δῖς; lat. *bis*, ai. *dvis* ‘zweimal’

**uítsuōs* > **uísuōs* > att. ἴσος, ostion. ἴσος, kret. Fίσφος ‘vielleicht’

ee > *ē* bzw. *ē*

**é-deue* > ion-att. ἔδει ‘es musste’

oo > *ō* bzw. *ō*

**-oso* > ion-att. -ου, inseldor. -ω (G.Sg. o-St.)

54 **tš* **tš* scheint in einigen ionischen Dialekten länger erhalten geblieben zu sein, da ein extra Zeichen für diesen Laut benutzt wird, auch wenn die genaue Lautung unsicher ist. Daher setze ich diesen Lautwandel eher spät an (und ungenauerweise in allen Dialekten gleichzeitig). In den meisten Dialekten wird **tš* zu *ss*, das im Anlaut vereinfacht wird. Im Att., Euboi., Boiot. und Kret. wird es jedoch zu *tt* bzw. *t* im Anlaut.

tš > *t* \ #_ (att. euboi. boiot. kret.)

tš > *tt* (att. euboi. boiot. kret.)

tš > *s* \ #_

tš > *ss*

s. die Beispiele bei 21

55 Hauchdissimilation (GRASSMANN'S Gesetz) GRASSMANN'S Gesetz wird in der Literatur nie präzise formuliert und meist ungenau oder gar nicht datiert (Rix §107, Bartoněk 2003, S. 147, Lejeune 1972, §45, Sihler 1995, §138). Da die lautgesetzlichen Reflexe durch morphologischen Druck ausgeglichen werden konnten, ist die genaue Wirkungsweise auch nicht mehr mit Sicherheit festzustellen. Nach einer möglichen Formulierung wird ein aspirierter Silbenanlaut (**h* oder Tenuis aspirata) vor Tenuis aspirata in der folgenden Konsonantenverbindung deaspiriert. Da im In- und Auslaut meist Derivations- und Flexionsmorpheme stehen, die von dem Gesetz (vielleicht aber aus Systemzwang) nicht betroffen zu sein scheinen, beschränke ich meine Formulierung auf den Anlaut. Die Dissimilation **-t^hē-t^hi* > -θη-τι im Imp.2.Sg.Aor.Pass. wird von der obigen Formulierung nicht abgedeckt, da hier der zweite Laut deaspiriert wird.

Nach Bartoněk ist nicht klar, ob das Gesetz vor- oder nachmyk. ist, aber wie Lejeune mit ὄφρα < **ióp^hra* (cf. Stamm des Relativums **io-*) ‘bis; damit’ zeigt, muss es nach **iV-* > *hV-* (s. 29) datiert werden. In den Absätzen zu diesem Gesetz liefert Mayrhofer (1986, §4.10.2.1.3) sogar inschriftliche Belege für nicht-dissimilierte Wörter. Daher datiere ich dieses Gesetz erst spät.

$(T^b, h) > (T, \emptyset) \setminus \#_C^*VC^*T^b$

zu uridg. **seǵ^b-*: ἔχω ‘ich habe’ vs. ἔξω ‘ich werde haben’

τάφος ‘Grab’ vs. θάπτω ‘ich bestatte’

τριχός ‘des Haares’ vs. θρίξ ‘Haar’

nicht-dissimiliert: θυφλός statt τυφλός ‘Blind’, ἀνεθέθη statt ἀνετέθη ‘er weihte’ &c.

56 Psilose In einigen Dialekten geht anlautendes *h* verloren (die einzige Position, in der es überhaupt vorkommt). Die genaue Datierung (gerade bei später belegten Dialekten wie dem Lesb.) ist unsicher.

$h > \emptyset$ (el. lesb. ostion. inselion. kret.)

57 \bar{e} Das aus *ā* gehobene \bar{e} fällt mit $\bar{\epsilon}$ zusammen, ist jedoch im Inselion. noch länger erhalten, wo die beiden Laute graphisch unterschieden sind.

$\bar{e} > \bar{\epsilon}$ (inselion. später)

58 Kontraktionen Einige Kontraktionen passieren nach dem Digammaschwund (und vermutlich nach 57), sind aber ebenso schwer genau zu formulieren wie die bei 49 übersprungenen und werden deswegen ebenfalls übergangen.

6. Schluss und Ausblick

Die Arbeit hat gezeigt, wie nützlich ein Programm wie *lga* bei der Sortierung und Formulierung von Lautgesetzen ist. Viele Fehler, die einem dabei passieren, können oftmals schnell aufgedeckt werden und die Gnadenlosigkeit eines Computerprogramms zwingt einen zu Exaktheit und Strenge. Diese sind (sofern sie so gut es geht der Realität entsprechen wollen) allerdings nur auf Basis einer vernünftigen Beleglage möglich; die Literatur führt jedoch Formen oft ohne Belegstellen an, wodurch sie schwer überprüfbar sind (Fundort und Datierung wären z.B. oft wünschenswert zu wissen) und belegen außerdem nie alle beschriebenen Fälle. Die Beispiele, die ich bei der Beschreibung der Lautgesetze gegeben habe, sollten also in Zukunft zu wirklichen Belegen werden.

Offensichtlich wurden in dieser Arbeit viele Lautentwicklungen noch völlig ignoriert oder noch nicht angemessen beschrieben. Besonders die Veränderungen ab dem frühen 1. Jt., in dem sich die Dialekte stark differenzieren, sind schwer in den Griff zu bekommen. Sie bietet aber immerhin schon ein Gerüst für weitere Forschung bzw. die Einarbeitung des in der weitergehenden Literatur detaillierter beschriebenen Lautwandels.

Obwohl die Resultate automatisch erzeugt wurden, erfolgte die Überprüfung immer noch manuell. Auch hier ist noch methodischer Verbesserungsbedarf. In der Softwareentwicklung bedient man sich heutzutage meist Unittests, mit denen man automatisch überprüfen kann, ob bei einer Reihe von Testfällen für eine bestimmte Eingabe die zu erwartende Ausgabe herauskommt. Da wir es allerdings aufgrund der vielen Dialekte nicht mit einer, sondern vielen Ausgaben zu tun haben, wird die Situation komplexer bzw. unübersichtlicher und für eine neue Version von `lga` sollte man sich über eine elegante Lösung Gedanken machen.

Neben methodischen sind aber auch technische Unzulänglichkeiten deutlich geworden. Eine sinnvolle Behandlung von Digraphen (bzw. Polygraphen) ist nötig. Außerdem ist die Art, auf die Laut- bzw. Zeichenklassen implementiert sind, weder konzeptionell sauber, da man sie im Laufe der Sprachentwicklung nicht ändern kann, noch performant, da jeder reguläre Ausdruck vorbearbeitet werden muss. Das erste Problem kann man, wie ja schon erwähnt, lösen, indem man Polygraphen in jeder Zeichenkette (d.h. regulären Ausdrücken und Eingabewörtern) durch ansonsten unbenutzte Unicode Codepoints ersetzt und vor der Ausgabe wieder zurückersetzt. Das zweite Problem ist mit einer bestehenden Regex-Engine nicht zu lösen; hier ist eine eigene Implementation mit einer eigenen Sprache nötig, die auf die Bedürfnisse der Lautgesetzformulierung optimiert ist. Ein Feature, das so eine Lautgesetzsprache haben sollte, wäre eine Möglichkeit, Akzente wenn nötig zu ignorieren, da ein regulärer Ausdruck `/ab/` z.B. nicht `"áb"` matcht und man so zu teilweise sehr umständlichen Ausdrücken gezwungen ist. Für einige Sprachen (für das Griechische allerdings eher weniger) wäre auch die Möglichkeit, mit Silben zu arbeiten, nützlich. Darüber, wie die Beschreibungssprache genau aussehen sollte und welche weiteren Features wünschenswert, notwendig und vernünftig implementierbar sind, wird man sich noch Gedanken machen müssen. Für die meisten Fälle funktioniert `irregex` schon erstaunlich gut, aber die Lesbarkeit ist (wie oft bei regulären Ausdrücken) nicht optimal.

Ein weiterer Punkt ist die Anwendung der Regellisten, um eine phonetische und graphematische Form zu erhalten. Diese müssen z.Z. noch für jede Sprache explizit vergeben werden, was unschön ist, aber nicht wichtig genug war, sofort verbessert zu werden.

Was die Benutzung betrifft ist `lga` noch völlig unintuitiv; die nächste Version sollte eine graphische Benutzeroberfläche haben. Lautgesetze sollten kommentierbar und mit Literaturverweisen und Belegen versehen werden können. Um die Sortierung zu erleichtern, wäre es vielleicht nützlich, Lautgesetze mit Beschränkungen zu versehen, die verhindern, dass zwei Lautgesetze, deren relative Chronologie zueinander klar ist, nicht fälschlicherweise in die falsche Reihenfolge kommen. Dass einem bei kompli-

zierteren Fällen, in denen mehrere Lautwandel ineinandergreifen, solche Fehler schnell passieren, hat die Erfahrung gezeigt.

Schließlich ist es mein Wunsch, mit *lga* und seinen zukünftigen Nachfolgern die historisch-vergleichende Sprachwissenschaft mehr wie freie Softwareentwicklung zu betreiben und (hoffentlich) als Gemeinschaftsprojekt eine Sammlung aller Lautgesetze zu erstellen, die die Lautentwicklung vom Urindogermanischen in die Einzelsprachen möglichst gut modellieren. Das Projekt soll als dann Referenz und Grundlage für weitere Forschung dienen können. Aktuell findet sich sowohl die Arbeit als auch *lga* samt den bisher eingetragenen Lautgesetzen unter <https://github.com/aap/lga>. Diese Arbeit ist nur der erste Schritt.

Literatur

- Allen, Sidney S. (1958). "Some problems of palatalization in Greek". In: *Lingua* 7, S. 113–33.
- Bartoněk, Antonín (1966). *Development of the long-vowel system in Ancient Greek dialects*. Prag: Státní pedagogické nakladatelství.
- (1991). *Grundzüge der altgriechischen mundartlichen Frühgeschichte*. Innsbruck.
- (2003). *Handbuch des mykenischen Griechisch*. Heidelberg: Winter.
- Beekes, Robert S. P. (2010). *Etymological Dictionary of Greek*. Leiden, Boston: Brill.
- Buck, Carl D. (1910). *Introduction to the study of the Greek dialects: grammar, selected inscriptions, glossary*. Boston: Ginn und Company.
- Katičić, Radoslav (1970). *A contribution to the general theory of comparative linguistics*. The Hague, Paris: Mouton.
- Lejeune, Michel (1972). *Phonétique historique du mycénien et du grec ancien*. Paris: Éditions Klincksieck.
- Lillo, Antonio (1992). "Zu griechisch τίπτει". In: *Glotta* 60, S. 15–19.
- Mayrhofer, Manfred und Warren Cowgill (1986). *Indogermanische Grammatik. Lautlehre. Segmentale Phonologie des Indogermanischen*. Bd. I/2. Heidelberg: Winter.
- Osthoff, Hermann und Karl Brugman (1878). *Morphologische Untersuchungen auf dem Gebiete der indogermanischen Sprachen*. Bd. I. Leipzig: Verlag von S. Hirzel.
- Rix, Helmut (1992). *Historische Grammatik des Griechischen*. 2., korrigierte Auflage. Darmstadt: Wissenschaftliche Buchgesellschaft.
- Sihler, Andrew L. (1995). *New Comparative Grammar of Greek and Latin*. Oxford/New York: Oxford University Press.
- Simkin, Oliver B. (2004). *Osthoff's Law: A Study in Greek Historical Phonology*. University of Cambridge: PhD. Dissertation.

Sommer, Ferdinand (1905). *Griechische Lautstudien*. Straßburg: Verlag von Karl J. Trübner.

Wackernagel, Jakob (1888). “Miscellen zur griechischen grammatik”. In: *Zeitschrift für vergleichende Sprachforschung* 29, S. 124–52.

A. Quelltext

Der Unterschied zwischen vorkombinierten Zeichen (also mit einem Codepoint) und kombinierten Zeichen (mehrere Codepoints) ist leider im Abdruck nicht mehr erkenntlich. Grundsätzlich wurden außer bei Akzenten wenn möglich vorkombinierte Zeichen verwendet. Auf <https://github.com/aap/lga> ist die korrekte und ausführbare digitale Form zu finden.

A.1. main.scm

```
;;;
;;; main file
;;;

(require-extension utf8)
(require-extension irregex)
(require-extension srfi-1)

(define s+ string-append)

;;;
;;; assoc lists
;;;

(define sound-classes '())
(define phonological-rules '())
(define prettyprint-rules '())

;; TODO: clean up the mess
;; Associates a name with a set of sounds
(define (set-class key value)
  (let ((x (assoc key sound-classes)))
    (if x
        (set-cdr! x value)
        (set! sound-classes (cons (cons key value) sound-classes)))))

;; Associates a name with a list of rules to get to phonetic representation
(define (set-phono-rules key value)
  (let ((x (assoc key phonological-rules)))
    (if x
        (set-cdr! x value)
        (set! phonological-rules (cons (cons key value) phonological-rules)))))

;; Associates a name with a list of rules to pretty-print words
(define (set-pretty-rules key value)
  (let ((x (assoc key prettyprint-rules)))
```

```

    (if x
      (set-cdr! x value)
      (set! prettyprint-rules (cons (cons key value) prettyprint-rules))))

(define (lookup key alist)
  (let ((x (assoc key alist)))
    (if x
      (cdr x)
      '())))

(define (l key)
  (lookup key sound-classes))

;;;
;;; tree and rule functions
;;;

;; makes a list of all languages between start and end language based on a tree
(define (make-path tree start end)
  (define (find-start tree)
    (cond ((atom? tree)
          (if (eq? tree start)
              tree
              '()))
          ((eq? (car tree) start)
           tree)
          (else (let loop ((t (cdr tree)))
                  (if (null? t)
                      '()
                      (let ((s (find-start (car t))))
                        (if (null? s)
                            (loop (cdr t))
                            s)))))))
    )
  (define (to-end tree)
    (cond ((atom? tree)
          (if (eq? tree end)
              (cons tree '())
              '()))
          ((eq? (car tree) end)
           (cons end '()))
          (else (let loop ((t tree))
                  (if (null? t)
                      '()
                      (let ((p (to-end (car t))))
                        (if (null? p)
                            (loop (cdr t))
                            (cons (car tree) p)))))))
    )
  (to-end (find-start tree)))

;; make a list of rules that apply to the language development in path
(define (make-rules tree path)
  (define start (car path))
  (define end (car (reverse path)))
  (define (rule-applies? langlist)
    (or (null? langlist)

```

```

        (if (equal? (car langlist) 'not)
            (eq? '() (lset-intersection eq? path (cdr langlist)))
            (not (eq? '() (lset-intersection eq? path langlist)))))
(define (find-start tree)
  (cond ((null? tree)
        '())
        ((eq? (car tree) start)
         (cdr tree))
        ((and (list? (car tree))
              (eq? 'br (caar tree)))
         (let ((l (find-start (caddar tree))))
           (if (null? l)
               (find-start (cdr tree))
               l)))
        (else (find-start (cdr tree)))))
(define (to-end tree)
  (cond ((or (null? tree)
            (eq? (car tree) end))
        '())
        ((procedure? (car tree))
         (cons (car tree) (to-end (cdr tree)))) ; better use tail recursion
        ((and (list? (car tree))
              (rule-applies? (cadar tree)))
         (case (caar tree)
             ((br) (to-end (caddar tree)))
             ((sub) (append (to-end (caddar tree)) (to-end (cdr tree))))
             (else (to-end (cdr tree)))))
        (else (to-end (cdr tree)))))
(to-end (find-start tree))

;;;
;;; helper functions
;;;

;; convert list of strings to single string with one original string per line
(define (make-words lst)
  (if (null? lst)
      '()
      (let ((s (make-words (cdr lst))))
        (if (null? s)
            (car lst)
            (s+ (car lst) "\n" s)))))

;; make regex substitute function
(define (s pattern . subst)
  (let ((pat (string->irregex (make-regex pattern) 'utf8 'fast 'm)))
    (lambda (word)
      (apply irregex-replace/all (cons pat (cons word subst))))))

;; compile regex containing custom character classes
(define (make-regex s)
  (let loop ((classes sound-classes)
            (s s))
    (if (null? classes)
        s
        (loop (cdr classes)
              (string-append s (make-regex (car classes)))))

```

```

(irregex-replace/all (s+ "<"
                        (symbol->string (caar classes))
                        ">")
                      s
                      (cdar classes))))))

;; apply a list of rules to a string, return transformed string
(define (apply-rules rules words)
  (if (null? rules)
      words
      (apply-rules (cdr rules)
                    ((car rules) words))))

;; runs 'words' from language 'from' through language 'to'
(define (run-list from to words)
  (let ((r1 (make-rules rules (make-path lang-tree from to)))
        (r2 (lookup from phonological-rules))
        (r3 (lookup to prettyprint-rules)))
    (apply-rules r3
                  (apply-rules r1
                                (apply-rules r2 (make-words words))))))

;;;
;;; data
;;;

;; tree of languages
;; an atom is a leaf language
;; a list's car is a language, the cdr its children
(define lang-tree
  '(uridg (urgr1 (urgr (nwgr (dor (nwdor phok lokr ait1)
                                     (sardor kor meg oarg)
                                     (sdor el lak inseldor warg kret))
                                (aiol boiot (thess wthess othess) lesb))
              (sogr (ach myk ark kypr pamph)
                    (ion-att (ostion hom) inselion euboi att))))))

))

;; sound classes
;; TODO: these should be changeable by the rules
(set-class 'kurz-vok "e|E|o|O|a|A|i|u|ə")
(set-class 'lang-vok "ē|Ē|ō|Ō|ā|Ā|ī|ū")
(set-class 'vok (s+ (l 'kurz-vok) "|" (l 'lang-vok)))
(set-class 'liquid "r|ɾ|l|ł")
(set-class 's-liquid "R|L")
(set-class 'nasal "m|n|ɳ|ñ|ń|ň")
(set-class 's-nasal "M|N")
(set-class 'halb-vok "y|w")
(set-class 'res (s+ (l 'liquid) "|" (l 'nasal)))
(set-class 'res-yw (s+ (l 'res) "|" (l 'halb-vok)))
(set-class 's-res (s+ (l 's-liquid) "|" (l 's-nasal)))
(set-class 'lary "H1|H2|H3|H")
(set-class 'sibil "s|š|ş|ś|z!|z|ž!|ž|z!|z")
(set-class 'labial "b!|p!|b|p")
(set-class 'dental "d!|t!|d|t")
(set-class 'retroflex "ɖ!|ʈ!|ɖ|ʈ")

```

```

(set-class 'palatal "j!|č!|j|č")
(set-class 'palatal2 "j!|c!|j|c")
(set-class 'velar "g!|k!|g|k")
(set-class 'labiovelar "G!|K!|G|K")
(set-class 'okklu (s+ (l 'labial) "|" (l 'dental) "|" (l 'retroflex)
                    "|" (l 'palatal) "|" (l 'palatal2) "|" (l 'velar)
                    "|" (l 'labiovelar)))
(set-class 'kons (s+ (l 'res) "|" (l 'halb-vok) "|" (l 'sibil) "|"
                    (l 'okklu) "|" (l 'lary)))
(set-class 'media "b|d|đ|j|j|g|G")
(set-class 'tenuis "p|t|ť|č|c|k|K")
(set-class 'mediaasp "b!|d!|đ!|j!|j!|g!|G!")
(set-class 'tenuisasp "p!|t!|ť!|č!|c!|k!|K!")

(load "subfuncs.scm")
(load "rules.scm")

```

A.2. subfuncs.scm

```

;;;
;;; functions that transform submatches
;;;

;; applies list of rules to the submatch i in a match
(define (match-rulelist rules)
  (lambda (i)
    (lambda (match)
      (let ((m (or (irregex-match-substring match i)
                    "")))
        (apply-rules rules m))))))

(define (match-do f)
  (lambda (i)
    (lambda (match)
      (let ((m (irregex-match-substring match i)))
        (f m)))))

(define match-print
  (match-do
   (lambda (s)
     (display " ")
     (display s)
     (newline)
     s)))

(define laryngal->vokal
  (match-rulelist
   (list (s "H1" "e")
         (s "H2" "a")
         (s "H3" "o")
         (s "H" "a"))))

(define laryngal->langvokal
  (match-rulelist
   (list (s "H1" "ē")
         (s "H2" "ā"))))

```

```

        (s "H3" "ō")
        (s "H" "ā"))))

(define dehnung
  (match-rulelist
    (list (s "a" "ā")
          (s "e" "ē")
          (s "o" "ō")
          (s "i" "ī")
          (s "u" "ū"))))

(define dehnung2
  (match-rulelist
    (list (s "a" "ā")
          (s "e" "Ē")
          (s "o" "Ō")
          (s "i" "ī")
          (s "u" "ū"))))

(define kuerzung
  (match-rulelist
    (list (s "ā" "a")
          (s "ē|Ē" "e")
          (s "ō|Ō" "o")
          (s "ī" "i")
          (s "ū" "u"))))

(define dental->retroflex
  (match-rulelist
    (list (s "t" "ṭ")
          (s "d" "ḍ"))))

(define palatal->velar
  (match-rulelist
    (list (s "č" "k")
          (s "j" "g"))))

(define labiovelar->velar
  (match-rulelist
    (list (s "K" "k")
          (s "G" "g"))))

(define labiovelar->labial
  (match-rulelist
    (list (s "K" "p")
          (s "G" "b"))))

(define labiovelar->dental
  (match-rulelist
    (list (s "K" "t")
          (s "G" "d"))))

(define velar->labiovelar
  (match-rulelist
    (list (s "k" "K")
          (s "g" "G"))))

```



```

(define sonans->consonans
  (match-rulelist
    (list (s "i" "y")
          (s "u" "w")
          (s "R" "r")
          (s "L" "l")
          (s "N" "n")
          (s "M" "m"))))

(define consonans->sonans
  (match-rulelist
    (list (s "y" "i")
          (s "w" "u")
          (s "r" "R")
          (s "l" "L")
          (s "n" "N")
          (s "m" "M"))))

(define stimmhaft
  (match-rulelist
    (list (s "K" "G")
          (s "k" "g")
          (s "č" "j")
          (s "c" "j")
          (s "t" "d")
          (s "ṭ" "ḍ")
          (s "p" "b")
          (s "s" "z")
          (s "š" "ž"))))

(define stimmlos
  (match-rulelist
    (list (s "G" "K")
          (s "g" "k")
          (s "j" "č")
          (s "j" "c")
          (s "d" "t")
          (s "ḍ" "ṭ")
          (s "b" "p")
          (s "z" "s")
          (s "ž" "š"))))

(define ->F
  (match-rulelist
    (list (s "k(!?)" "χ")
          (s "g(!?)" "γ")
          (s "t(!?)" "θ")
          (s "d(!?)" "δ")
          (s "p(!?)" "f")
          (s "b(!?)" "β"))))

(define depala
  (match-rulelist
    (list (s "w̃w̃" "w")
          (s "r̃r̃" "r"))

```

```

(s "{l" "l")
(s "ńń" "n"))))

(define ->acute
  (match-rulelist
    (list (s "~" ""))))

(define ->circumflex
  (match-rulelist
    (list (s "" "~"))))

(define ->A
  (match-rulelist
    (list (s "(!|$)" "!"))))

(define ->!A
  (match-rulelist
    (list (s "!"))))

(define ->M
  (match-rulelist
    (list (s "(.)" (->!A 1))
          (s "(.)" (stimmhaft 1)))))

(define ->MA
  (match-rulelist
    (list (s "(.)" (->A 1))
          (s "(.)" (stimmhaft 1)))))

(define ->T
  (match-rulelist
    (list (s "(.)" (->!A 1))
          (s "(.)" (stimmlos 1)))))

(define ->TA
  (match-rulelist
    (list (s "(.)" (->A 1))
          (s "(.)" (stimmlos 1)))))

```

A.3. rules.scm

Diese Datei definiert den Lautgesetzbaum und die phonologischen und graphematischen Regeln. Alle anderen Zweige außer dem Griechischen sind hier nicht abgedruckt.

```

(load "griechisch.scm")

(define rules
  (list
    'uridg
    '(br (urgr1) ,rules-urgr)
  ))

(set-phono-rules 'uridg
  (list (s "h" "!")
        (s "kw" "K")
        (s "gw" "G")

```

```

(s "ĥ" "č")
(s "ĝ" "ĵ")
(s "ĥ" "y")
(s "ŭ" "w")
(s "Ṁ" "M")
(s "Ṇ" "N")
(s "Ṛ" "R")
(s "Ṭ" "L")
(s "h1" "H1")
(s "h2" "H2")
(s "h3" "H3")
(s "\u0301" "'") ; combining acute
;;;
(s "e('?)H2" "a" 1 "H2")
(s "H2e" "H2a")
;;; umgefärbtes e noch von o verschieden
(s "e('?)H3" "0" 1 "H3")
(s "H3e" "H30")
;;; TODO: bedingungen für stimmhaftwerdung
(s "s(G|g|ĵ|d|b|r|l|m|n)" "z" 1)
(s "(G|g|ĵ|d|b|r|l|m|n)(!?)s" 1 2 "z")
;;; TT -> TsT
(s "(<dental>(<dental>)" 1 "s" 2)
)

(let ((rules (list (s "h" "h")
                  (s "kw" "K")
                  (s "gw" "G")
                  (s "ĥ" "y")
                  (s "ŭ" "w")
                  (s "\u0301" "'")))))
  (set-phono-rules 'urgr1 rules)
  (set-phono-rules 'urgr rules))

(let ((rules (list (s "!" "h")
                  (s "G" "gw")
                  (s "K" "kw")
                  (s "w" "ŭ")
                  (s "y" "ĥ")
                  (s "ā" "ā")
                  (s "ē" "ē")
                  (s "ī" "ī")
                  (s "ō" "ō")
                  (s "ū" "ū")
                  (s "Ā" "ā")
                  (s "Ē" "ē")
                  (s "Ō" "ō")
                  (s "~" "\u0303") ; combining tidle
                  (s "'" "\u0301")))) ; combining acute
  (set-pretty-rules 'urgr1 rules)
  (set-pretty-rules 'urgr rules)
  (set-pretty-rules 'lesb rules)
  (set-pretty-rules 'ion-att rules)
  (set-pretty-rules 'ostion rules)
  (set-pretty-rules 'inselion rules)
  (set-pretty-rules 'hom rules))

```

```

(set-pretty-rules 'euboi rules)
(set-pretty-rules 'att rules)
(set-pretty-rules 'kret rules)
(set-pretty-rules 'dor rules)
(set-pretty-rules 'sdor rules)
(set-pretty-rules 'sardor rules)
(set-pretty-rules 'inseldor rules)
(set-pretty-rules 'kypr rules)
(set-pretty-rules 'thess rules)
(set-pretty-rules 'boiot rules)
(set-pretty-rules 'el rules)
(set-pretty-rules 'kor rules)
(set-pretty-rules 'warg rules)
(set-pretty-rules 'oarg rules)
(set-pretty-rules 'ark rules)
)

; Set to #t to have Greek printed out in Greek alphabet
(define usegrkalphabet #f)

(define (togrk str)
  (define rules (list
    (s "υ̣υ̣" "υ̣")
    (s "ι̣ι̣" "ι̣")
    (s "\u0301" "'") ; combining acute

    (s "a(['~]?)ι̣" "αι" 1)
    (s "a" "α")
    (s "ā(['~]?)ι̣" "α" 1 "ι")
    (s "ā" "α")
    (s "e(['~]?)ι̣" "ει" 1)
    (s "e" "ε")
    (s "ē" "ει")
    (s "ē(['~]?)ι̣" "η" 1 "ι")
    (s "ē" "η")
    (s "o(['~]?)ι̣" "οι" 1)
    (s "o" "ο")
    (s "ō" "ο")
    (s "ō(['~]?)ι̣" "ω" 1 "ι")
    (s "ō" "ω")
    (s "u(['~]?)ι̣" "υι" 1)
    (s "υ" "φ")
    (s "u" "υ")
    (s "ū" "υ")
    (s "i" "ι")
    (s "ī" "ι")

    (s "b" "β")
    (s "g" "γ")
    (s "d" "δ")
    (s "sd" "ζ")
    (s "tʰ" "θ")
    (s "k" "κ")
    (s "l" "λ")
    (s "m" "μ")
    (s "n" "ν")
  ))

```

```

(s "ks" "ξ")
(s "p" "π")
(s "r" "ρ")
(s "s$" "ς")
(s "s" "σ")
(s "t" "τ")
(s "ph" "φ")
(s "kh" "χ")
(s "ps" "ψ")
(s "' " "\u0301") ; combining acute
(s "~" "\u0303") ; combining tilde
))

(if usegrkalphabet
  (apply-rules rules str)
  str))

```

A.4. griechisch.scm

Diese Datei enthält sämtliche für das Griechische definierte Lautgesetze inklusive stichwortartiger Notizen.

```

(define rules-urgr
  (list
    ;;; Ein Laut der Einfachheit halber
    (s "z" "s")

    ;;; *e neben *H3 und *o fallen zusammen
    (s "0" "o")

    ;;; Kentum -- Rix §92-94
    (s "č" "k")
    (s "j" "g")

    ;;; Thorn -- Rix §91 -- Schindler 'A thorny problem'
    (s "(k|K)þ" 1 "t")
    (s "(g!|G!)þ" 1 "d!")

    ;;; MA > TA -- Rix §94
    (s "<mediaasp>" (stimmlos 1))

    ;;; (H)ῑ im Anlaut -- gegen Rix §68,80e
    (s "^y" "dy")
    (s "^(<lary>)y" "y")

    ;;; Laryngale -- Rix §79-85
    ; Rix §85d
    (s "<kons>(H1|H2)$" 1 (laryngal->vokal 2))
    ; Rix §85
    (s "<kons>(<kons>)(i|u)(H1|H2)$"
      1 2 3 (laryngal->vokal 4))
    (s "(i|u)(H1|H2)$" (sonans->consonans 1) (laryngal->vokal 2))
    ; Dehnung -- Rix §82b,84,85a
    (s "<vok>('?)(<lary>)(<kons>|<s-res>|$)"
      (dehnung 1) 2 (sonans->consonans 4))
    ; Rix §80d, §82c

```

```

(s "(<kons>|^)(<lary>)(<kons>)" 1 (laryngal->vokal 2) 3)
; Rix §85c
(s "NH2$" "na")
; Rix §84
(s "(<kons>)(<lary>)(<s-res>)" 1 (laryngal->vokal 2) (sonans->consonans 3))
; Rix §83e
(s "(<s-res>)(<lary>)(<kons>)"
  (sonans->consonans 1) (laryngal->langvokal 2) 3)
(s "(<s-res>)'(<lary>)(<kons>)"
  (laryngal->vokal 2) "" (sonans->consonans 1) (laryngal->vokal 2) 3)
; Rix §79bc
(s "^(<lary>)(<s-res>)" (laryngal->vokal 1) (sonans->consonans 2))
; kompletter Verlust -- Rix §79a,81a,82d,84
(s "<lary>")

;;; vor Labiovelar > Velar wegen εἴnov -- nicht bei Rix
(s "wew" "wey")

;;; Labiovelar > Velar -- Rix §97
(s "(w|u|ū)('?)(<labiovelar>)" 1 2 (labiovelar->velar 3))
(s "(<labiovelar>)(w|u|ū|y)" (labiovelar->velar 1) 2)

;;; silbische Resonanten
; Rix §76
(s "(<s-nasal>)(<vok>|<halb-vok>)" "a" (sonans->consonans 1) 2)
(s "(<s-nasal>)" "ə")
; Rix §75
(s "(<s-liquid>)(\$|<vok>|<halb-vok>)" "ə" (sonans->consonans 1) 2)
(s "(<s-liquid>)" (sonans->consonans 1) "ə")

;;; Assimilationen (nach Labiovelarwandel noch mal!) -- Rix §78
(s "(<nasal>)(<labial>)" "m" 2)
; gegen Rix nicht *ms > *ns wegen ἔνειμα < *enemsa (oder analog?)
(s "(<nasal>)(<dental>|<velar>|<labiovelar>)" "n" 2)
; Rix §77
(s "my" "ny")

;;; 1. Kontraktion (nach Laryngalverlust)
; TODO: Langvokale? gibt es hier überhaupt welche?
; Rix §81
(s "ee" "ē")
(s "e'e" "ē~")
(s "aa" "ā")
(s "a'a" "ā~")
(s "oo" "ō")
(s "o'o" "ō~")
; i,u > y,w
(s "(i|u)('?)(<vok>)" 1 2 (sonans->consonans 1) 3)
(s "(<vok>)(['~]?)(i|u)" 1 2 (sonans->consonans 3))

;;; unmittelbare Assimilation -- Rix §106a
; TODO: früher (idg); phonologisch?
(s "(<okklu>)(<media>)([!])" (->M 1) 2 3)
(s "(<okklu>)(<tenuis>)([!])" (->T 1) 2 3)
(s "(<okklu>)(<tenuisasp>)([!])" (->TA 1) 2 3)

```

```

;; Kw > Kw (Beibehaltung des Silbengewichts)
(s "(<vok>)([ '~ ]?)(<velar>w)" 1 2 (velar->labiovelar 3) (velar->labiovelar 3))
(s "(<velar>w)" (velar->labiovelar 1))
(s "(<velar>)(<labiovelar>)" 2 2)

;; Beseitigung von Geminaten (αἰπόλος < *aig-kwolos) -- nicht Rix
(s "(<kurz-vok>)([ '~ ]?)(<kons>)" 1 2 3 "#")
(s "GG" "G") (s "KK" "K") (s "K!K!" "K!")
(s "gg" "g") (s "kk" "k") (s "k!k!" "k!")
(s "bb" "b") (s "pp" "p") (s "p!p!" "p!")
(s "dd" "d") (s "tt" "t") (s "t!t!" "t!")
(s "#")

;;; Okklusive neben s -- Rix §105
;; Assimilation über s; Tenuis vor s
(s "(<media>)s(<tenuis>)([ ^! ])" 1 "s" (stimmhaft 2) 3)
(s "(<tenuisasp>)s(<tenuis>)([ ^! ])" 1 "s" (->A 2) 3)
(s "(<okklu>)s" (->T 1) "s")

; CsC -- Rix §87c
; s zwischen gleichen Konsonanten -- TODO: alle Fälle abgedeckt?
(s "ks(<velar>)" "s" 1)
(s "Ks(<labiovelar>)" "s" 1)
(s "ts(<dental>)" "s" 1)
(s "ps(<labial>)" "s" 1)
(s "rsr" "sr") (s "lsl" "sl")

; vor s > h und Palatalisierung
'urgr1

;;; s > h
; s > h \ ($|V)_(V|R) -- Rix §88,89
(s "(^|(<vok>)([ '~ ]?))s(<vok>|<res>|w)" 1 "h" 4)
;; VRsV > VRhV in unbetonten Silben (Datierung?)
(s "(<vok>)(<res>|w)s(<vok>)" 1 2 "h" 3)
;; VNsV > VNhV
(s "(<vok>)([ '~ ]?)(<nasal>)s(<vok>)" 1 2 3 "h" 4)

;; ab hier substrat-s

; sy,ys > yy -- Rix §89g, Lejeune §127
; TODO: Datierung
; was neben Konsonant (Csy)? vermutlich verlust
; was im Anlaut sy- (ὄμῆν)
(s "(<vok>)([ '~ ]?)sy" 1 2 "yy")
(s "^sy" "y")
; TODO: Beispiele (-oῖσι > myk. -o-i)
(s "ys(<vok>)" "yy" 1)

;;; Osthoff -- Rix §58,64
; Wohl nach *ns > *nh wegen μῆνός < *mēnsós und
; vor Verlust von Okklusiven im Auslaut
; *nh (und *nm, *uῖ?) nicht betroffen
(s "nh" "#nh")
(s "(<lang-vok>)([ '~ ]?)(<res-yw>)(<kons>)" (kuerzung 1) 2 3 4)
(s "#")

```

```

;;; Konsonanten im Auslaut (Datierung unklar) -- Rix §100
; nach *r̥C > *rəC wegen ὑπόδρα < *upo-dr̥k
(s "(<okkl̥u>)+$")
; Rix §77
(s "m$" "n")

;;; Palatalisierung -- Rix §102
(s "^(t!|t)y" "s")
(s "(t!|t)y" "ts")
(s "(d|g)y" "j̥j")
(s "(k!|k)y" "čč")
(s "(<labial>)y" "pč")

(s "ly" "l̥l")
(s "ry" "r̥r")
(s "ny" "n̥n")

;;; Datierung? -- Rix §105
(s "dl" "ll")

;;; tk, tkw, tp Metathese -- Rix §106 (etwas anders)
(s "(t)(k|K|p)" 2 1)

;;; Datierung unklar, vermutlich später und gestaffelt -- Rix §78
(s "^m(<liquid>)" "b" 1)
(s "m(<liquid>)" "bm" 1)
(s "^n(<liquid>)" "d" 1)
(s "n(<liquid>)" "nd" 1)

;; u- > hu-
; datierung unklar
(s "^u" "hu")

;;
;; Urgriechisch
;;
'urgr

; ti > si -- Rix §101
; Kein wirkliches (ausnahmsloses) Lautgesetz;
; schwierig einigermaßen zufriedenstellend zu formulieren.
'(sub (sogr) (,(s "([̂sk])ti" 1 "si"))))

; TODO: wann o?
'(sub (aiol) (,(s "ə" "o"))))
(s "ə" "a")

(s "pč" "pt")

; heteromorphemisches ty
; TODO

(s "^y(<vok>)" "h" 1)
(s "(<vok>)([ '̃~]?)y(<vok>)" 1 2 "h" 3)

```



```

;;
;; Mykenisch
;;
'myk

(s "čč" "tš")
(s "ǰǰ" "ds")

'(sub (el) (,(s "ē" "Ā"))))

;;; Labiovelare > T/P -- Rix §96-99
'(sub (not aiol) (,(s "K(i|i)" "t" 1)))
'(sub (not kypr aiol) (,(s "<labiovelar>(e|ē)" (labiovelar->dental 1) 2)))
(s "<labiovelar>" (labiovelar->labial 1))
;; Assimilationen an Labiovelarreflexe
(s "<nasal>(<labial>)" "m" 2)
(s "<nasal>(<dental>)" "n" 2)
(s "<nasal>(<velar>)" "ŋ" 2)

;; b,g > m,ŋ \ _n -- Rix §105
(s "bn" "mn")
(s "gn" "ŋn")

;;; ca. 1000

;; tw -- Rix §104
'(sub (not kret) (,(s "^tw" "s"))))
(s "(.)tw" 1 "tš")
;; thw -- unklar, aber vgl. -σθε αι. -dhvam
(s "(.)t!w" 1 "st!")
;(s "^t!w" "t!")

;; ds > sd
; vor CsC > ChC wegen ἔρδω < *uerzdō < *uerĝie/o-
(s "ds" "sd")

;; (N|T)sC > sC -- Rix §78, Lejeune §134
; vor CsC > ChC wegen ἱσοο < ὑιδσυος und δεσπότης < *demspot-
; nach dz > zd wegen σύζυγος < *sun-dzugos
(s "<nasal>|<dental>)s(<kons>)" "s" 2)

;;; CsC -- Rix §87c, Lejeune §132-133
; TODO: πασταδ-/παρταδ- < parstad-
; θύσθεν < *thursthen
(s "<res-yw>s(<tenuis>|<tenuisasp>)" 1 "#s" 2)
(s "<kons>s(<kons>)" 1 "h" 2)
(s "<tenuis>h" (->A 1))
(s "<tenuisasp>(<tenuis>)([!])" 1 (->A 2) 3)
(s "<tenuisasp>(<media>)([!])" (->M 1) 2 3)
(s "#")

;;; ūī > w̄w̄ -- Rix §73
(s "wy" "w̄w̄")

;;; Palatale řř ľľ řń -- Rix §70
; TODO: Datierung? wohl nach myk.

```

```

(s "(a|o)([']~?)(řř|ńń|w̃w̃)" 1 2 "y" (depala 3))
'(sub (kypr) (,(s "a([']~?)ĺĺ" "a" 1 "yl"))))
(s "ĺĺ" "ll")
'(sub (thess lesb) (,(s "(řř|ńń)" (depala 1) (depala 1))))

;;; 1. Ersatzdehnung/Gemination
;; Palataldehnung -- Rix §73
(s "(e|i|u)([']~?)(řř|ńń|w̃w̃)" (dehnung2 1) 2 (depala 3))
;; Rh/hR-Dehnung
'(sub (not thess lesb) (,(s "<vok>([']~?)<res>|w)h<vok>"
                        (dehnung2 1) 2 3 4)
      ,(s "<vok>([']~?)h<res>|w)<vok>"
          (dehnung2 1) 2 3 4)))
;; ln-Dehnung
'(sub (not thess lesb) (,(s "<vok>([']~?)ln<vok>" (dehnung2 1) 2 "l" 3)))
(s "<vok>([']~?)ln<vok>" 1 2 "ll" 3)
;; 7 vs. 5 Langvokale
'(sub (not ion-att nwdor sardor) (,(s "Ē" "ē")
                                   ,(s "Ō" "ō"))))

;; ws > wh
;; TODO: dialektal rs > rh (oder später rs > rr ?)
(s "<vok>([']~)(w)s<vok>" 1 2 3 "h" 4)

;; Gemination und Hauchumsprung
(s "^<vok>([']~?)h<vok>([']~)" "h" 1 2 3 4)
(s "^<vok>([']~?)<res>|w)h" "h" 1 2 3 "h")
;; TODO: prohorā > φρourá aspirieren
(s "<vok>([']~?)<res>|w)h<vok>" 1 2 3 3 4)
(s "<vok>([']~?)h<res>|w)<vok>" 1 2 3 3 4)
;; h > θ (nicht im anlaut)
(s "(.)h" 1)

;;; Cts > Cs
(s "<kons>ts" 1 "s")
;;; ts > ss/tt -- Rix §102,87
'(sub (boiot kret) (,(s "ts" "tt"))))
(s "ts" "ss")
;; ss > s nach Langvokal
(s "<lang-vok>([']~?)ss" 1 2 "s")
'(sub (ion-att ark) (,(s "ss" "s"))))

;;; ds -- Rix §102
'(sub (boiot kret lak el) (,(s "sd" "dd"))))
(s "^dd" "d")

;;; ca. 900

;;; ion. ā > ǣ
'(sub (ion-att) (,(s "ā" "ǣ"))))

;;; ca. 800

;;; 2. Ersatzdehnung/Diphthongierung
;; TODO: bei allen Vokale gleich? (nicht im Kyrenischen)
;; VnsV

```

```

;; TODO: dialekte? sardor, arg, achaisch?
'(sub (lesb) (,(s "(<vok>)([ '~]? )ns(<vok>)" 1 2 "ys" 3)))
'(sub (ion-att boiot el lak inseldor kypr)
      (,(s "(<vok>)([ '~]? )ns(<vok>)" (dehnung2 1) 2 "s" 3)))
;; Vns$
'(sub (lesb el) (,(s "(<vok>)([ '~]? )ns$" 1 2 "ys")))
'(sub (ion-att boiot lak inseldor)
      (,(s "(<vok>)([ '~]? )ns$" (dehnung2 1) 2 "s")))
;; Vokaleinreihung
;; TODO: dialekte? inseldorisch einteilung (ost west)? thera?
'(sub (not ion-att nwdor inseldor) (,(s "Ē" "ē"
                                         ,(s "ō" "ō"))))

;;; ältere Kontraktionen, TODO: datieren
;; wegen Unsicherheit vorläufig auskommentiert
; a, ā + e
; '(sub (ion-att) (,(s "(a|ā)'e" "ā~")
;                  ,(s "(a|ā)e" "ā"))))
; (s "(a|ā)'e" "ē~")
; (s "(a|ā)e" "ē")
; ; a+0
; (s "a'(o|ō|ō)" "ō~")
; (s "a(o|ō|ō)" "ō")
; ; att. a+e
; '(sub (att) (,(s "e'a" "ē~")
;                ,(s "ea" "ē"))))
; ; att. e+o,ō
; '(sub (att) (,(s "e'o" "ō~")
;                ,(s "e'ō" "ō~")
;                ,(s "eo([ ^'])" "ō" 1)
;                ,(s "eō([ ^'])" "ō" 1))))
; ; e + ē
; (s "e'ē" "ē~")
; (s "eē" "ē")

;;; w > θ
; TODO: dialekte. vieles unklar
; V_V früh -- Buck §53
'(sub (ion-att inseldor lesb kret ark) (,(s "(<vok>)([ '~]? )w(<vok>)" 1 2 3)))
; _V später (TODO: wann > h?)
'(sub (att) (,(s "^w(<vok>)([ '~]? )s(<kons>)" "h" 1 2 "s" 3)))
'(sub (ion-att inseldor lesb) (,(s "^w(<vok>)" 1)
                                ,(s "(<vok>)([ '~]? )yw(<vok>)" 1 2 "yy" 3)
                                ,(s "^hw" "h"))))
'(sub (ion-att inseldor) (,(s "^wr" "hr"))))

;;; hR-
;; TODO: dialekte
'(sub (ion-att) (,(s "^h(l|m|n)" 1)))

;;; att. æ > ā \ eir_
'(sub (att) (,(s "(((e|ē|Ē|i|ī)([ '~]? ))|r|y)Ā" 1 "ā"))))

;;; ca. 700

```

```

;;; 3. Ersatzdehnung/Schwund; e+e, o+o Kontraktion
;; TODO: dialekte nach Rix §72; Bartoněk, Buck?
; [wo]arg?
; sw kret. erhalten
'(sub (inselion ostion inseldor kret) (,(s "^(^|(<vok>)([']~?))(r|l|n|d|t!|s)w" (dehnung2 1) 4)))
'(sub (ion-att lesb lak) (,(s "(r|l|n|d|t!|s)w" 1)))
; Kontraktion
(s "e'e" "Ē~")
(s "ee" "Ē")
(s "o'o" "Ō~")
(s "oo" "Ō")
;; TODO: dialekte?
'(sub (not ion-att nwdor sardor) (,(s "Ē" "ē")
                                   ,(s "Ō" "ō"))))

'(sub (att euboi boiot kret) (,(s "tš" "tt")))
(s "^tt" "t")
(s "tš" "ss")
(s "^ss" "s")

;; Graßmann -- Rix §107
(s "<tenuisasp>(<kons>)*(<vok>)([']~?)((<kons>)*(<tenuisasp>)"
  (->T 1) 2 4 5 6 8)
(s "h(<kons>)*(<vok>)([']~?)((<kons>)*(<tenuisasp>)"
  1 3 4 5 7)

;; Psilose -- Rix §68
'(sub (el lesb ostion inselion kret) (,(s "h"))))

;;; ca. homerisch; Sprachstufe der ionischen Archaismen
'hom

'(sub (not el inselion) (,(s "Ā" "ē"))))

;; jüngere Kontraktionen; teilweise nach Einsetzen der Überlieferung
;; wegen Unsicherheit vorläufig auskommentiert
; ; o + a > ō (nach w > θ)
; (s "o'a" "ō~")
; (s "oa" "ō")
; ; ē + o,ō; TODO: genauere Bedingungen
; '(sub (att) (,(s "ē'ō" "ō~"))))
; (s "ē([']~?)(o|ō)" "e" (->acute 1) "ō")
; ; e + ē (nach ion. ā > ē)
; (s "e'ē" "ē~")
; (s "eē" "ē")
; ; ā + o,ō; TODO: datierung? dialekte (aiol, hom)?
; '(sub (dor) (,(s "ā([']~])ō" "ā~")
;               ,(s "āō" "ā"))))
; '(sub (dor) (,(s "ā([']~])o" "ā~")
;               ,(s "([']~])(.*)āo" 1 2 "ā")
;               ,(s "āo([']~)" "ā" 1)))
;
))

```

A.5. run-gr.scm

Diese Datei ist mit "csi run-gr.scm" auszuführen und führt die Simulation an den entsprechenden Wörter aus.

```
(load "main.scm")

(define run
  (lambda () (load "run-gr.scm")))

(define (run-grassmann dest)
  (print " " (symbol->string dest))
  (print (run-list 'urgr1 dest (list
    "sékhō"
    "tháphiō"
    "tháphos"
    "thíthēmi"
    "thrikhós"
    "syéthos"
    "iōphra"
  )))
  (newline))
(run-grassmann 'att)

(define (run-contr dest)
  (print " " (symbol->string dest))
  (print (run-list 'urgr1 dest (list
    "enīkae"
    "nīkái1"
    "timáomen"
    " ́āo"
    "iā~uos"
    "lāuós"
    "nasuós"
    "au1sós"
    "lāuokrínēs"
    "-āsōm"
    "theāuōrós"
    "poteidāuōn"
    "koināuōn"
    "u1étesa"
    "kréuas"
    "-éās"
    "génesos"
    "thesós"
    "suādéuos"
    "philéiō nt i"
    "philéioi"
    "-ē~os"
    "hikwkwóuanaks"
  )))
  (newline))
;(run-contr 'att)
;(run-contr 'ostion)
;(run-contr 'hom)
;(run-contr 'lesb)
;(run-contr 'boiot)
```

```

;(run-contr 'kret)

(define (run-misc dest)
  (print " " (symbol->string dest))
  (print (run-list 'uridg dest (list
    "uēuṛh₁mh₁nos"
    "uṛh₁mṇ"
    "h₂ensīēh₂"
    "demspot-"
    "uṛidsuōs"
    "orsmā"
    "h₂eūsriom"
    "uērgīoh₂"
    "ph₂ṛtīh₂"
    "kʷelesīetaī"
    "gelosīos"
  )))
  (print (run-list 'urgr1 dest (list
    "plangiō"
    "sīumēn"
  )))
  (print (run-list 'urgr dest (list
    "sunjjugos"
  )))
  (newline))
(run-misc 'urgr)
(run-misc 'att)
(run-misc 'oarg)

(define (run-y dest)
  (print " " (symbol->string dest))
  (print (togrk (run-list 'urgr1 dest (list
    "káuṛiō"
    "auṛietós"
    "glukéuṛia"
    "eṛréuṛia"
    "díuṛios"
    "élaīuom"
    "eláīuā"
    "aiṛési"
    "póīuā"
  ))))
  (newline))
(run-y 'urgr)
(run-y 'att)
(run-y 'lesb)

(define (run-labvel dest)
  (print " " (symbol->string dest))
  (print (run-list 'uridg dest (list
    "kʷis"
    "kʷe"
    "kʷidkʷe"
    "gʷoukʷolos"
    "gʷih₃os"
    "h₁lṅgʰus"
  )))

```

```

    "kwukwlos"
    "penkwe"
    "gwelbhus"
    "gwenhioh2"
    "gwhhonos"
    "gwṁioh2"
    "kwoineh2"
    "kwinuoh2"
    "snigwḥṁ"
    "ogwḥis"
    "penkwtos"
    "kwḷHim"
    "-gwḥnet"
    "sens"
  )))
  (newline))
(run-labvel 'urgr1)
(run-labvel 'urgr)
(run-labvel 'att)
(run-labvel 'lesb)
(run-labvel 'ark)

(define (run-ky dest)
  (print " " (symbol->string dest))
  (print (run-list 'urgr1 dest (list
    "kḷāmeron"
    "phulákḷō"
    "glōkḷia"
    "tṽéḷisō"
    "thuariós"
  )))
  (print (run-list 'uridg dest (list
    "kwétṽres"
    "-dhṽe"
  )))
  (newline))
(run-ky 'ostion)
(run-ky 'euboi)
(run-ky 'att)
(run-ky 'boiot)
(run-ky 'kret)
(run-ky 'sdor)

(define (run-s-test dest)
  (print " " (symbol->string dest))
  (print (run-list 'urgr1 dest (list
    "asa"
    "a'sa"
    "asra"
    "a'sra"
    "asla"
    "a'sla"
    "asma"
    "a'sma"
    "asna"
    "a'sna"
  )))

```

```

"aswa"
"a'swa"
"sa"
"sra"
"sla"
"sma"
"sna"
"swa"
)))
(newline))
(run-s-test 'urgr)
(run-s-test 'att)

(define (run-2ed dest)
  (print " " (symbol->string dest))
  (print (run-list 'urgr1 dest (list
    "-ántia"
    "-óntia"
    "-éntia"
    "-áns"
    "-óns"
    "-éns"
  )))
  (newline))
(run-2ed 'urgr)
(run-2ed 'lesb)
(run-2ed 'thess)
(run-2ed 'boiot)
(run-2ed 'ostion)
(run-2ed 'el)
(run-2ed 'ark)
(run-2ed 'kret)
(run-2ed 'inseldor)

(define (run-wau dest)
  (print " " (symbol->string dest))
  (print (run-list 'urgr1 dest (list
    "kalṽós"
    "ksénṽos"
    "kórṽā"
    "néṽā"
    "khōrá"
    "dyeṽínós"
    "dédṽimen"
  )))
  (newline))
(run-wau 'att)
(run-wau 'ostion)
(run-wau 'inseldor)
(run-wau 'kret)
(run-wau 'el)
(run-wau 'kor)
(run-wau 'ark)
(run-wau 'boiot)
(run-wau 'lesb)

```



```

(define (run-s dest)
  (print " " (symbol->string dest))
  (print (run-list 'urgr1 dest (list
    "ploksmós"
    "ekstrós"
    "aĩksmā"
    "persnā"
    "orsmā"
    "parstādes"
    "thúrsthen"
    "prosorā"
    "akóysō"

    "éysō"
    "éserpon"
    "iserós"

    "īsáomai"
    "isániō"
    "esús"
    "īsātro-"
    "witswos"
  )))
  (newline))
(run-s 'urgr)
(run-s 'att)
(run-s 'lesb)

```

```

(define (run-led dest)
  (print " " (symbol->string dest))
  (print (run-list 'urgr1 dest (list
    "korsā"
    "akoysā"
    "aysōs"
    "oysatós"
    "kórsā"
    "akóysō"
    "éysō"
    "aysiō"
    "khésras"
    " selásnā"
    "khánsas"
    "énemsa"
    "kwélson"
    "nasuós"
    "gwolnā"
    "mēnsós"
    "mēńs"
    "yésma"
  )))
  (newline))
(run-led 'urgr)
(run-led 'att)
(run-led 'inseldor)
(run-led 'kret)
(run-led 'lesb)

```

```

(define (run-pala dest)
  (print " " (symbol->string dest))
  (print (run-list 'uridg dest (list
    "gwῆιοhz"
    "κοῖιος"
    "αἰῖος"
  )))
  (print (run-list 'urgr1 dest (list
    "καhαριῖο"
    "πhτηριῖο"
    "οἰκτιριῖο"
    "κτηνιῖο"
    "κρινιῖο"
    "πλυνιῖο"
  )))
  (newline))
(run-pala 'ion-att)
(run-pala 'kypr)
(run-pala 'thess)
(run-pala 'boiot)

(define (run-myk)
  (let ((dest 'ion-att))
    (print (run-list 'uridg dest (list
      "Ἡῖος"
      "τρεῖς"
      "δhτις"
      "δι῔ς"
      "δι῔μ"
      "τοτῖος"
      "μεdhῖος"
      "ποdσι"
      "μττός"
      "ἡ2ῥός"
    )))
    (run-list 'urgr dest (list
      "pantsa"
    )))
  (run-myk))

(define (run-gr-kw dest)
  (print " " (symbol->string dest))
  (print (run-list 'uridg dest (list
    "αἰgκwόλος"
    "ἡῖέκυος"
  )))
  (newline))
(run-gr-kw 'urgr)
(run-gr-kw 'att)

(define (run-gr-s)
  (print (run-list 'uridg 'att (list
    "αὔςιῖο"
    "psde-"
    "πρδhskoh2"
  )))

```

"dikskos"
"leg^hskeh₂"
"mĩgskoh₂"
"ũoidt^hh₂e"
"ũidd^hi"
"parstádes"
"t^húrst^hen"
"kenttós"
"tũe-"
"etũe-"
)))
(run-gr-s)