

Morphologische Analyse mit Toolbox

Karl Heinz Wagner[†]

1. Einleitung

Dieses Dokument zeigt, wie die Lexika für die morphologische Analyse (*morphological parsing*) mit Toolbox aufgebaut sein müssen, damit die gewünschten Resultate erzielt werden können. Das Programm ermöglicht eine weitgehend automatisierte morphologische Analyse, wenn in den Lexika die dafür erforderliche morphologische Information in geeigneter Form enthalten ist. Wie im Folgenden gezeigt werden wird, ist das Programm sogar in der Lage, morphophonemische (oder morphographemische: fly+s → flies) Veränderungen zu erfassen, die bei der Verknüpfung von Wurzeln und Affixen vielfach auftreten können (s.u. Abschnitt 5.).

Zum leichteren Verständnis werden nach Möglichkeit englische Beispiele oder solche aus bekannteren Sprachen herangezogen.

2. Einfache Analyse: Wurzeln und Affixe

Eine Voraussetzung für die korrekte Analyse ist, dass im Lexikon die Morpheme im Lexemfeld (\lx) entweder als Wurzeln oder als Affixe (Präfixe, Suffixe, Infixe) ausgewiesen sind. Dafür müssen die Affixe mit einer geeigneten Markierung der Morphemgrenze (*morpheme break character*) versehen werden. Das ist standardmäßig der Bindestrich.¹ Diese Einstellung kann jedoch im Programm geändert werden. Grundsätzlich gelten folgende Regeln:

- Ein Präfix muss mit einem Bindestrich abschließen: *un-*
- Ein Suffix muss mit einem Bindestrich beginnen: *-ful*
- Ein Infix muss in Bindestrichen eingeschlossen sein: *-um-*
- Eine Wurzel darf keine Bindestriche aufweisen, auch wenn die Wurzel nur gebunden vorkommt: *book*

Dies ist alles, was für die einfache Analyse benötigt wird. Mit den Lexikoneinträgen

\lx un-	\lx success	\lx -ful
\ps neg	\ps N	\ps adj
\gl anti	\gl Erfolg	\gl voll_von

wird *unsuccessful* richtig wie folgt interlinearisiert:

unsuccessful		
un-	success	-ful
anti-	Erfolg	-adj
neg-	N	-voll_von

Wortformen, die bereits einen Bindestrich aufweisen, werden ebenfalls richtig analysiert. Mit den Einträgen

\lx non-	\lx read	\lx -er
\ps neg	\ps V	\ps nom
\gl nicht	\gl les	\gl Agens

im Lexikon wird *non-reader* wie folgt analysiert:

non-reader		
non-	read	-er
nicht-	les	-Agens
neg-	V	-nom

[†] Stark überarbeitete und eingedeutschte Fassung von David Bevan, *Parsing with Shoebox* aus der Shoebox Dokumentation

¹ Es ist unbedingt darauf zu achten, dass zwischen dem Bindestrich und dem Affix **kein** Leerraum steht.

2.1 Infixe

Ein Infix ist ein Morphem, das in eine Wurzel oder in ein anderes Affix eingefügt wird. Das darf nicht mit einem Morphem verwechselt werden, das zwischen einer Wurzel und einem weiteren Affix steht. In diesem Fall handelt es sich nur um ein weiteres Präfix oder Suffix. In einer Form wie *readers* (Plural von *reader*) {read}{-er}{-s} ist *-er* keinesfalls ein Infix, sondern ein Suffix, auf das ein weiteres Suffix folgt.

Ein Beispiel für ein **echtes** Infix findet sich im Tagalog beispielsweise in einer Wortform wie *bumili* 'kaufte' mit der Wurzel *bili* 'kaufen' und dem Infix *-um-* 'Agens'.

$$\begin{array}{c} \text{kaufte} \\ \underbrace{\hspace{1.5cm}} \\ b - \underbrace{\text{um}} - \text{ili} \\ \underbrace{\hspace{1.5cm}} \\ \text{Agens} \\ \underbrace{\hspace{1.5cm}} \\ \text{kaufen} \end{array}$$
 Mit den Lexikoneinträgen

\lx -um- \ps infix \gl Agens	\lx bili \ps V \gl kauf
------------------------------------	-------------------------------

wird die Tagalog-Wortform *bumili* korrekt wie folgt analysiert:

bumili	
bili	-um-
kauf	-infix-
V	-Agens-

In der Analyse-Ausgabe stehen die Infixe wahlweise entweder vor oder nach der Wurzel oder dem Stamm in dem Sie vorkommen. Dies kann im Toolbox-Programm eingestellt werden.

Auch im Englischen findet sich wortspielerisch Infigierung in allen möglichen Formen, z.B. in *absobloominlutely* aus *absolute-ly* + *-bloomin-*, *viomalin* aus *violin* + *-ma-*, *fanfuckingtastic* aus *fantastic* + *-fucking-*. Mit den Lexikoneinträgen

\lx absolute \ps A \gl absolute	\lx -ly \ps suff \gl adv	\lx -bloomin- \ps infix \gl intens
---------------------------------------	--------------------------------	--

erhalten wir folgende Analyse:

absobloominlutely		
absolute	-bloomin-	-ly
absolut	-intens-	-adv
A	-infix-	-suff

2.2 Zusammengesetzte Wurzeln (Komposita)

Wörter mit zusammengesetzten Wurzeln werden richtig analysiert, wenn beide Wurzeln im Lexikon stehen. Mit den Lexikoneinträgen

\lx black \ps A \gl schwarz	\lx bird \ps N \gl Vogel	\lx -s \ps suff \gl Pl
-----------------------------------	--------------------------------	------------------------------

wird das Kompositum *blackbirds* folgendermaßen interlinearisiert:

blackbirds			
black	-	bird	-s
schwarz	-	Vogel	-Pl
A	-	N	-suff

Komposita wie *sea-green* und *mother-in-law*, die bereits Bindestriche enthalten, werden richtig analysiert. Im Toolbox-Programm kann man einstellen, ob zusammengesetzte Wurzeln zugelassen werden sollen oder nicht.

3. Alternative Formen

Wenn ein Morphem mehr als seine Oberflächenform aufweist, kann dies in einem eigenen Feld für alternative Formen angegeben werden, die mit der Markierung `\a` gekennzeichnet wird. Der englische unbestimmte Artikel *a* z.B. hat eine alternative Form *an*, die vor vokalisch anlautenden Wörtern verwendet wird. Für das Wort *telephone* gibt es eine Kurzform *phone*.

```
\lx a           \lx telephone
\la an         \la phone
\ps Art        \ps N
\gl indef      \gl Telefon
```

In die Analyse-Ausgabe wird die Form aus dem Lexemfeld `\lx` übernommen:

an	enormous	phone
a	enormous	telephone
indef	sehr_gross	Telefon
Art	A	N

Es folgen einige weitere Beispiele von Lexikoneinträgen für Morpheme mit alternativen Formen:

```
\lx in-         \lx -s           \lx not
\la im-         \la -es          \la -n't
\la il-         \ps suff         \ps neg
\la ir-         \gl Pl          \gl nicht
\ps neg
\gl anti
```

Interlinearisierter Beispieltext:

impossible	foxes	faces	haven't
in- possible	fox -s	face -s	have -not
anti- möglich	Fuchs -Pl	Gesicht -Pl	hab nicht
neg- A	N	-suff N	-suff V neg

Die Formen *foxes* und *faces* werden richtig analysiert, wenn die Wurzeln *fox* und *face* im Lexikon vorhanden sind. Man beachte, dass *haven't* als Kompositum analysiert worden ist und nicht als Folge von separaten Wörtern.

Der Morphemtyp (Suffix, Präfix etc.) alternativer Formen kann vom Eintrag im Lexemfeld `\lx` abweichen. Eine Wurzel oder ein Präfix kann z.B. als alternative Form ein Suffix haben. In dem Esperanto-Ausdruck *La eta princo* 'der kleine Prinz' ist das Adjektiv *eta* aus dem Diminutivsuffix *-et* abgeleitet. Wir können das durch zwei verschiedene Lexikoneinträge erfassen, einen für das Suffix *-et* und einen anderen für die Wurzel *et*. Dabei würde aber nicht zum Ausdruck kommen, dass es sich um Varianten handelt. Statt dessen könnte das Suffix als Variante des Stammes betrachtet werden.

4. Zugrunde liegende Formen (*Underlying Forms*)

Wenn aus welchen Gründen auch immer eine Variante einen eigenen Lexikoneintrag haben soll, kann dafür ein Feld für eine "zugrunde liegende Form" (engl. *underlying form*) verwendet werden, die mit der Markierung `\u` gekennzeichnet wird. Dieser Eintrag verweist dann auf einen oder mehrere andere Lexikoneinträge mit spezifischen Informationen.

Im Folgenden finden Sie eine alternative Behandlung der Kurzform *phone* für *telephone*. Der Haupteintrag ist *telephone* während der Eintrag für *phone* quasi zu lesen ist als "siehe unter *telephone*".

```
\lx phone      \lx telephone
\u telephone   \ps N
               \gl Telefon
```

In dem Datenfeld für die zugrunde liegende Form (*underlying form*) kann auch vorgegeben werden, wie eine Form in Morpheme zerlegt werden muss. Dies bietet sich besonders für Fälle mit suppletiven Formen an, bei welchen der Zusammenhang mit einer Wurzel nicht mehr erkennbar ist. Ein gutes Beispiel dafür ist das engl. *went* als Vergangenheitsform von *go*. Diese Form stammt aus dem Paradigma eines anderen Verbs, nämlich *wend* (vgl. dt. *winden*). Die Angaben in dem `\u`-Feld `go -ed` verweisen jeweils auf den Eintrag für *go* und *-ed*.

Zugrunde liegende Formen (Underlying Forms)

\lx went	\lx go	\lx -ed
\u go -ed	\ps V	\ps suff
	\gl geh	\gl past

Wichtig!! Zwischen den Formen in einem \u-Feld muss ein Zwischenraum stehen, damit Wurzeln, Präfixe und Suffixe deutlich unterschieden werden können. In der Online-Hilfe zu Toolbox wird dies leider falsch dargestellt.

Hier ist das Ergebnis:

he	went		
he	go	-ed	
3SM	geh	-past	
Pro	V	-suff	

Statt nun aber für jede unregelmäßige Form einen eigenen Lexikoneintrag zu machen, kann man diese in die Einträge für die Wurzeln als alternative Formen mit jeweils zugeordneter zugrunde liegender Form angeben:

\lx go	\lx find	\lx hit
\a went	\a found	\u hit
\u go -ed	\u find -ed	\u hit -ed
\ps V	\ps V	\ps V
\gl geh	\ge find	\gl schlag

Jedes \u-Feld für eine zugrunde liegende Form bezieht sich auf das vorangehende \lx- oder \a-Feld.

Die Verbform *hit* ist mehrdeutig (Präsensstamm oder Vergangenheitsform), weshalb hier zwei Angaben zur zugrunde liegenden Form gemacht werden. Bei der Anwendung wird das Programm vom Benutzer eine Entscheidung anfordern, welche der Alternativen zu wählen ist. Beispiel:

went		found		hit	
go	-ed	find	-ed	hit	-ed
geh	-past	find	-past	schlag	-past
V	-suff	V	-suff	V	-suff

Für Toolbox spielt es also keine Rolle, ob eine Form mit spezifischer Information für die Analyse als ein separater Lexikoneintrag oder als eine alternative Form eines anderen Eintrags behandelt wird.

Gelegentlich kann es erforderlich sein, auch für komplexe Affixe Angaben zur zugrunde liegenden Form zu machen. Zwischen *readable* und *readability* besteht offensichtlich ein Zusammenhang und wir würden gerne ausdrücken, dass *-ability* in *readability* sich aus dem *-able* von *readable* und einem Suffix *-ity* zusammensetzt. Das Folgende zeigt die Lösung:

\lx -able	\lx -ity	\lx -ability
\ps suff	\ps suff	\u -able -ity
\gl bar	\gl nom	

Damit wird *readability* so analysiert:

readability			
read	-able	-ity	
les	-bar	-nom	
V	-suff	-suff	

Es folgen Beispiele für die zugrunde liegenden Formen von Komposita:

\lx have	\lx brunch
\a I've	\u breakfast lunch
\u I have	
\ps V	
\gl hab	

Das liefert folgende Analyse:

I've	brunch		
I	have	breakfast	lunch
1S	hab	Frühstück	Mittagessen
Pro	V	N	N

4.1 Erzwungene Werte — Auflösung von Ambiguitäten

Wenn ein Morphem mehrdeutig ist, kann es gelegentlich hilfreich sein, wenn in den zugrunde liegenden Formen spezifiziert werden kann, welche Bedeutung verlangt ist, so dass Toolbox keinen Dialog zur Auflösung der Ambiguität führen muss. Im Englischen ist das Suffix *-s* mehrdeutig. Es repräsentiert einerseits die 3. Person Präsens in der Verbalflexion (z.B. *(he) sing-s*) andererseits den Plural in der Nominalflexion (z.B. *dog-s*). Bei der Darstellung einer unregelmäßigen Form wie *men* als zugrunde liegender Form *man -s* ist klar, dass es sich um die Pluralform handeln muss. Dies kann dadurch sichergestellt werden, dass in geschweiften Klammern die richtige Lesart hinzugefügt wird. *man -s {Pl}*.

<code>\lx -s</code>	<code>\lx -s</code>	<code>\lx man</code>
<code>\a -es</code>	<code>\a -es</code>	<code>\a men</code>
<code>\ps suff</code>	<code>\ps suff</code>	<code>\u man -s{Pl}</code>
<code>\gl Pl</code>	<code>\gl 3S</code>	<code>\ps N</code>
		<code>\gl Mann</code>

Die Form *men* wird damit klaglos wie folgt analysiert:

men	
man	-s
Mann	-Pl
N	-suff

Damit erzwungene Glossen so funktionieren können, ist es notwendig, dass in der Ausgabe die Zeile mit den Glossen (hier: `\g`) unmittelbar auf die Zeile mit der Morphemanalyse (`\m`) folgt. Wenn die *part of speech*-Zeile oberhalb der Glossenzeile stünde, müsste die Disambiguierung über eine Spezifizierung wie `-s{suff}{Pl}` erfolgen.

Im folgenden Beispiel liegt eine Mehrdeutigkeit in der Wurzel vor. Das Wort *bear* kann einmal das Substantiv *bear* 'Bär' sein oder die Vergangenheitsform des Verbs *bear* 'tragen'.

<code>\lx bear</code>	<code>\lx bear</code>
<code>\ps N</code>	<code>\a bore</code>
<code>\gl Bär</code>	<code>\u bear{trag} -ed</code>
	<code>\ps V</code>
	<code>\gl trag</code>

Mit diesen Lexikoneinträgen kann *bore* wie unten gezeigt analysiert werden. (Allerdings ist *bore* selbst mehrdeutig, es kann auch 'bohren' bedeuten).

bore	
bear	-ed
trag	-past
V	-suff

4.2 Direkte Analyse — Methoden zur Vermeidung falscher Analysen

Mit den Lexikoneinträgen,

<code>\lx hop</code>	<code>\lx hope</code>	<code>\lx -s</code>
<code>\ps V ; N</code>	<code>\ps V</code>	<code>\a -es</code>
<code>\gl hüpf</code>	<code>\gl hoff</code>	<code>\ps suff</code>
		<code>\gl 3S</code>

wird das Wort *hopes* fälschlicherweise wie folgt analysiert:

hopes	
*hop	-s
*hüpf	-3S
V	-suff

Bei der Analyse löst Toolbox eine mögliche Ambiguität dadurch auf, dass es das längste Affix abtrennt. Damit wird verhindert, dass der Benutzer vom Programm mit vielen – zumeist falschen – Alternativen konfrontiert wird.

Im Falle von *hopes* bedeutet das, dass das Suffix *-es* abgetrennt wird, was natürlich zu einem falschen Ergebnis führt. Wie lässt sich das verhindern?

Um in derartigen Situationen die korrekte Analyse zu erzwingen, muss sie als zugrunde liegende Form vorgegeben werden.

```
\x hope
\A hopes
\u hope -s{3S}
\ps V
\gl hoff
```

Damit wird für *hopes* die folgende Analyse praktisch erzwungen:

hopes
hope -s
hoff -3S
V -suff

Hier ein weiteres Beispiel. Angenommen wir hätten Einträge für ein Präfix *demi-* 'halb' und ein Nomen *sting* 'Stich (z.B. einer Biene)'. Da Toolbox immer die längstmöglichen Wurzeln oder Affixe abtrennt würde ein Wort wie *demisting* (eigentlich so etwas wie "entnebeln") fälschlicherweise wie folgt analysiert werden:

demisting
*demi- sting
*halb- Stich
*präf- N

Gemeint ist aber eine Zerlegung von *demisting* in $[[de- mist] -ing]$. Wir erreichen dies durch Einträge für *demist* und *mist* (sowie *-ing*) wie folgt:

```
\x demist          \x mist
\u de- mist{Nebel}{V} \ps N ; V
                    \gl Nebel
```

Das Wort wird dann richtig analysiert als:

demisting
de- mist -ing
ent- nebel -nom
präf- V -suff

In manchen Fällen gibt es für eine Wortform mehr als eine richtige Analyse. Das englische *does* kann einerseits eine Flexionsform des Verbs *do* sein andererseits die Pluralform des Substantivs *doe* 'Reh'. In solchen Fällen benötigen wir separate Lexikoneinträge mit alternativen Formen und jeweils unterschiedlichen zugrunde liegenden Formen.

```
\x do          \x doe
\A does        \A does
\u do -s{3S}   \u doe -s{Pl}
\ps V          \ps N
\gl mach       \gl Reh
```

Toolbox wird in diesen Fällen den Benutzer auffordern, eine Entscheidung zu treffen, welcher Fall vorliegt.

does	does
do -s	doe -s
mach -3S	female_deer -Pl
V -suff	N -suff

5. Morphophonologie – Morphophonemik (engl. morphophonemics)

Die Morphophonologie befasst sich mit den systematischen phonologischen Prozessen, die bei der Kombination von Morphemen zu Wortformen auftreten. Toolbox bietet einige Möglichkeiten, morphophonologische Erscheinungen zu erfassen, wobei natürlich alternative Formen ($\backslash a$) und zugrunde liegende Formen ($\backslash u$) eine ganz entscheidende Rolle spielen. Die folgenden bekannten Beispiele aus dem Englischen sind zwar ORTHOGRAPHISCHE Alternationen, sie können aber dennoch als Musterbeispiele für die Behandlung der Morphophonologie mit Toolbox dienen.

Im Englischen gibt es einige orthographische Regeln für die Bildung der Flexionsformen der Verben und Substantive wenn die Stämme (bzw. Wurzeln) auf bestimmte Grapheme enden. In unseren bisherigen Beispielen haben wir die Endung für die 3. Person Singular und den regelmäßigen Plural bei den Nomina mit *-s* repräsentiert und die Endung für die Formen der Vergangenheit (einschließlich Partizip) mit *-ed*. Wenn die Endung *-s* an den Stamm *try* 'Versuch (N); versuchen(V)' antritt, muss das *y* durch *ie* ersetzt werden: *tried*. Wenn andererseits die Endung *-ed* angehängt wird, wird das *y* durch *i* ersetzt.

Etwas formaler dargestellt:

$y \rightarrow i / _ _ +ed$ *oder* $y + ed \rightarrow ied$
 $y \rightarrow ie / _ _ +s$ *oder* $y + s \rightarrow ies$

In Toolbox kann dies wie folgt umgesetzt werden:

\lx -ed	\lx -s
\a -d	\a -es
\a -ied	\a -ies
\u y+ed	\u y+s
\ps suff	\ps suff
\gl past	\gl 3S

Das funktioniert folgendermaßen: Das \a-Feld enthält die Oberflächenform des Suffixes *inklusive des geänderten Teils des Stammes* (der Wurzel) (m.a.W. eine Form wie *tried* wird in *tr-ied* aufgespalten). Das \u-Feld enthält den sich ändernden Teil des Stammes (der Wurzel) zuzüglich der zugrunde liegenden Form des Suffixes (also *-ed*). Diese beiden Teile müssen durch ein Morphemgrennzeichen (normalerweise +, ohne Leerzeichen!) voneinander getrennt werden, im vorliegenden Fall erhalten wir *y+ed*.

Im Folgenden können Sie sehen, wie die Formen von *try* und *tie* analysiert werden.

tried		tied		tries		ties	
try	-ed	tie	-ed	try	-s	tie	-s
versuch	-past	bind	-past	versuch	-3S	bind	-3S
V	-suff	V	-suff	V	-suff	V	-suff

Die Alternativform *-d* (für *-ed*) wird für Stämme benötigt, die bereits auf *-e* enden, z.B. *singe* 'versengen' – *singe -d*, die Alternativform *-es* (für *-s*) ebenfalls für bereits auf *-e* ausgehende Stämme und darüber hinaus für alle Fälle, in welchen der Stamm auf einen Sibilanten auslautet: *wish - wish-es*, *kiss - kiss-es*.

Eine weitere orthographische Erscheinung bei der Bildung der Vergangenheitsform ist die Konsonantenverdoppelung, z.B. *hop - hopped*, die man ähnlich behandeln könnte.

\lx -ed
\a -d
\a -pped
\u p+ed
\ps suff
\gl past

Damit ergibt sich folgende Analyse für *hopped*:

hopped
hop -ed
hüpf -past
V -suff

Das Unschöne an diesem Beispiel ist, dass man für jeden Konsonanten einen eigenen Eintrag benötigen würde und eine Generalisierung nicht möglich scheint.

Das koreanische Akkusativsuffix (Markierung für das direkte Objekt) hat zwei Allomorphe: {/lul/, /ul/}:

<i>ton</i>	'money'	<i>ton-ul</i>	'money-ACC'
<i>chayk</i>	'book'	<i>chayk-ul</i>	'book-ACC'
<i>tali</i>	'leg'	<i>tali-lul</i>	'leg-ACC'
<i>sakwa</i>	'apple'	<i>sakwa-lul</i>	'apple-ACC'

Offensichtlich wird *-ul* verwendet, wenn der Stamm auf einen Konsonanten endet und *-lul* nach einem Vokal. Natürlich bietet es sich hier an, die eine Form als Alternativform zur anderen zu analysieren. Aber um welchen phonologischen Prozess handelt es sich? Es könnte sich entweder um eine Tilgung handeln: ...C+-lul → ...C+

Reduplikation

Angenommen die Allomorphe des Präfixes *in-* seien wie folgt spezifiziert:

```
\lx in-
\la im-
\la il-
\la ir-
\ps neg
\gl anti
```

Unter dieser Voraussetzung würden Wörter wie *image*, *imam*, *iris* und *iron* falsch analysiert werden, wenn für sie nicht bereits ein eigener Lexikoneintrag existiert, z.B.:

iron	
*in-	on
*anti-	auf
*neg-	P

Wenn wir hingegen zugrunde liegende Formen formulieren, die den nachfolgenden Kontext berücksichtigen, können diese fehlerhaften Analysen vermieden werden:

```
\lx in-
\la imb-
\lu in+b
\la imm-
\lu in+m
\la imp-
\lu in+p
\la ill-
\lu in+l
\la irr-
\lu in+r
\ps neg
\gl anti
```

5.2 Ambiguitäten erkennen

Wenn ein Wort auf *-es* endet, können zwei Fälle vorliegen: Entweder wir haben es mit einem Wort zu tun, dessen Stamm oder Wurzel auf *-e* endet, an das das Affix *-s* angefügt wird, oder es handelt sich um das Affix *-es*. Da Toolbox im Zweifelsfall immer das längste Affix abtrennt, muss auf geeignete Weise Vorsorge getroffen werden, dass die Ambiguität auch erkannt wird, z.B. auf folgende Weise:

```
\lx -s
\la -es
\la -es
\lu e+s
\ps suff
\gl 3S
```

6. Reduplikation

Unter Reduplikation versteht man eine morphologische Operation, durch die ein Teil einer Wurzel oder eines Stammes (einer Basis) kopiert wird und mit oder ohne weiteres phonologisches Material entweder als Präfix oder als Suffix an die Basis angefügt wird, um grammatische Information auszudrücken. Es ist auch möglich, dass die ganze Basis redupliziert wird.

Man betrachte z.B. folgende Formen der Ponape-Sprache:²

<i>duhp</i>	'dive'	<i>du-duhp</i>	'be diving'
<i>mihk</i>	'suck'	<i>mi-mihk</i>	'be sucking'
<i>wehk</i>	'confess'	<i>we-wehk</i>	'be confessing'

Hier werden die 'durative' Verbformen dadurch gebildet, dass eine Folge CV- als Präfix redupliziert wird. Solche einfache reduplikative Prozesse können in Toolbox dargestellt werden. Dies geschieht durch einen Lexikoneintrag

² Beispiele aus Haspelmath (2002:24).

Reduplikation

für das Muster der reduplizierten Silbe (Reduplikant). Im Lexemfeld \lx steht ein Name für dieses Muster, der wie ein Affix verwendet wird und aus der Zeichenfolge “dup” und einem Hinweis auf die Struktur des Reduplikanten besteht, z.B. CVdup- für ein Präfix oder -dupCV für ein Suffix. Die Strukturmuster selbst stehen in den Feldern für alternative Formen, d.h. den \a-Feldern und bestehen aus Kombinationen von Variablen für Zeichenklassen wie [cons] und [vowel], die in der jeweiligen Sprachkodierung definiert werden.³ Für unser Ponape-Beispiel sähe der Lexikoneintrag wie folgt aus:

```
\lx dupCV-
\ a [cons][vowel]-
\ps pre
\gl durativ
```

Die oben aufgeführten Ponape-Formen würden dadurch folgendermaßen analysiert:

duduhp		mimihk		wewehk	
dupCV-	duhp	dupCV-	mihk	dupCV-	wehk
durativ-	dive	durativ-	suck	durativ-	confess
pre-	V	pre-	V	pre-	V

Als weiteres Beispiel wollen wir uns die folgenden Somali-Formen ansehen:⁴

<i>buug</i>	'book'	<i>buug-ag</i>	'books'
<i>fool</i>	'face'	<i>fool-al</i>	'faces'
<i>koob</i>	'cup'	<i>koob-ab</i>	'cups'
<i>jid</i>	'street'	<i>jid-ad</i>	'streets'

Die Pluralform wird hier dadurch gebildet, dass der Auslautkonsonant zusammen mit einem davorstehenden Vokal *-a* wiederholt wird. Das Muster ist also -dupaC.

```
\lx -dupaC
\ a -a[cons]
\ps suff
\gl pl
```

Die Analyse sieht damit wie folgt aus:

buugag		foolal		koobab	
buug	-dupaC	fool	-dupaC	koob	-dupaC
book	-pl	face	-pl	cup	-pl
N	-suff	N	-suff	N	-suff

Die Reduplikation einer ganzen Einheit findet sich in vielen Pidgin-Varianten:

<i>kos</i>	'curse'	<i>koskos</i>	'curse again and again'
<i>krai</i>	'cry'	<i>kraikrai</i>	'cry continuously'
<i>fain</i>	'lovely'	<i>fainfain</i>	'very lovely'

Sie kann in Toolbox wie folgt spezifiziert werden:

```
\lx dup
\ a [...]
\ps redup
\gl intens
```

koskos		kraikrai		fainfain				
dup	-	kos	dup	-	krai	dup	-	fain
intens	-	curse	intens	-	cry	intens	-	lovely
redup	-	V	redup	-	V	redup	-	A

³ Diese Variablen können auch anders heißen, z.B. K und V, oder C und V. Dies ist ein Frage der Definition in der Sprachkodierung. In der Standard-Sprachkodierung (d.h. *default.lng*) gibt es die Variablen *cons*, *vowel*, und *nasal*.

⁴ Beispiele aus Haspelmath (2002:24).