1. GETTING STARTED

To run Kimmo give the command RUN <LI.LIB> KIMMO in the EXEC mode. At this point the system will respond with the following message:

You are now in LISP. The following files have been loaded from <LI.LIB>: INIT.LISP, KIMMO.COM, XEQFNS.COM, MERGEFNS.COMS, AUTFNS.COM, DICFNS.COM. Do (KIMMO) to start up KIMMO. -- (KIMMO 'X) starts it up in language X.

After you do (KIMMO) or (KIMMO LANGUAGE) the top-level menu and option prompt will be displayed as follows (the default language is set to be ENGLISH):

** KIMMO TOP-LEVEL MENU **

current language: ENGLISH
current dictionary file: ENGLISHDIC.LISP
current automata file: ENGLISHAUT.LISP
using multiple automata

COMMANDS:
Q: Quit KIMMO
F: File Change
G: Generate
A: Analyze
D: Dump
M: Manipulate data files
N: New language
C: Change mode (single vs. multiple automata)
T: Set or examine Tracepoints
S: Show this menu
?: Show available commands

Option (Q,F,G,A,D,M,N,C,T,S,?):
The default current file specifications are displayed in the top part of the screen, together with the current language. In the bottom part are brief descriptions of the available options, and their mnemonic letters. This top-level menu is not automatically displayed again after execution of every option, with the exception of the "New language" option, since the user is likely to want the results of the preceding option to remain on the screen for a few moments at least for examination.

Repeated use of the program will make users rapidly familiar with the simple mnemonics for the options, particularly since the option prompt itself contains a reminder. Should the menu screen be required it can be displayed by using the "S" option. However, the current implementation of KIMMO makes use of the ASKUSER facility of INTERLISP which greatly facilitates the prompt/response type of interaction. The mnemonic letter is automatically expanded to the name of the option and that option is set in operation as soon as the letter is typed. A reminder of what the letters stand for will be displayed if the user types "?", and any invalid response will be ignored. All options can be typed in upper or lower case.

All of the prompt/response interactions which arise in the course of a KIMMO session are handled by this facility if they have a predetermined set of valid responses (i.e. not prompts requiring the user to specify a new filename or word). Either the prompts display the mnemonic letters or, if the number of options is small their full names may be displayed. In the latter case the mnemonic letter is capitalized (almost invariably the initial letter anyway). Questions expecting "yes" or "no" answers will accept "y" or "n". If the user is in any doubt as to the appropriate responses he can type "?" to be reminded.

If the user decides not to choose any of the available options within one of the top-level options, a simple carriage return (\(<cr>\)) will redisplay the top-level options prompt, or an alternative within the same top-level option. To conclude a KIMMO session however, the user must explicitly type "Q" to quit, to avoid inadvertently quitting the program when alterations to the data files may not yet have been written to files.

Descriptions of the remaining options follow, under the respective names of the top-level options.
2. F : FILE CHANGE

This option allows the default filenames constructed by KIMMO to be replaced with ones specified by the user. In response to the "F" option the system will answer:

Current automata file is ENGLISHAUT.LISP
Change it ?

The response "N" will move the process on to the next file name change option; otherwise, specification of a filename is required.

name? Newfilename

After an appropriate filename is given, the system responses with:

Automata file changed.

This sequence is then repeated for the dictionary file.

3. G : GENERATION

Typing "G" causes KIMMO to check whether the executive functions file and the automata file associated with the current language have been loaded in the current KIMMO session and to invoke the generation functions in the executive functions file. If the files have not been loaded in the current Lisp session they will be loaded and the necessary structures compiled for use in generation (GMACHINE, RMACHINE). Loading messages will appear, and the appropriate bookkeeping information, as well as an indication of the duration of the compilation process, followed by the generator prompt:

Please type in one or more lexical forms :

If the files have been loaded in a previous KIMMO session (but this same Lisp session) then the file names alone will be displayed, and the generator prompt. If they were loaded earlier in the current session then the generator prompt alone will be displayed. The user then enters a lexical form or forms, for which a surface representation will be generated and displayed on the next line. E.g.
(English)
:flit+ing
    flitting

(Japanese)
:kak+ru
    kaku

(Rumanian)
:stea+@+a copil+ilor
    steaua
copiilor

(French)
:instituteur+e
    institutrice

If the generator is unable to produce a surface representation by applying the phonotactic rules of the automata, the unworkable lexical form will be echoed with a question mark:

:quidDity
    quidDity?

Responding to the generator prompt (:) with a carriage return alone will end the generation cycle and display the prompt for the top-level options again.

4. A : ANALYSIS

For analysis both the automata and lexicon files are required as well as the executive functions file, and when the analyzer is invoked by typing "A" in response to the top-level options prompt the same checking process as described under "G: Generation" is performed for all three, and appropriate action taken. The analyzer then responds with its prompt:

Please type in one or more surface forms :

Surface forms can then be entered and the analyzer will postulate their underlying representation using the automata and lexicon. When no appropriate underlying form can be found, either because the surface form does not conform to the phonotactic rules of the automata, or because its morphological elements cannot be identified in the lexicon, or both, the analyzer will echo the form which defies analysis, with a question mark:
:slithy-tovedom
  slithy-tovedom?

As with the generator, the analysis cycle can be terminated by responding to the analyzer's prompt with a carriage return alone, and the user will be returned to the top-level options prompt.

5. D : DUMP

This option displays a second menu detailing the various dump options available to extract information from the automata and lexicon files. Their titles should be self-explanatory:

** KIMMO DUMP MENU **

Current language: ENGLISH

INFORMATION TO BE FOUND (TYPE START OF KEYWORD):

ALP: Alphabet
ALT: Alternations
F: Feasible pairs
N: Names of automata
E: Entry from dictionary
L: Lexicon from dictionary
A: Automata
M: This menu
<cr>: Quit to top level
?: Show available commands

The dump option prompt will then be displayed:

Dump Option (ALP,ALT,F,N,E,L,AUT,M):

In the case of the lexicon entry and full lexicon dumps the user will receive a second prompt after responding with the option, indicating the current default lexicon and giving an opportunity to specify a different one:

Default is Root lexicon.
Display different lexicon? yes
Lexicon to be displayed:

The appropriate information is then displayed and the dump option prompt appears again in case more information is required. Note that when retrieving an entry from a suffix lexicon it should be specified exactly in the form in which it is stored, including any morpheme boundary marker such as '+':
Form for which to display entry: +ed
Default lexicon to search is Root lexicon.
Search different lexicon? yes
Lexicon to search: VERBENDINGS
Entry for '+ed' is: (# "PAST")
Dump Options (ALP,ALT,F,N,E,L,AUT,M):

A carriage return takes the user back to the top level:

Dump Options (ALP,ALT,F,N,E,L,AUT,M):
Quitting Dump...
Option (Q,F,G,A,D,M,N,T,S):

6. M : MANIPULATING DATA FILES

The automata and dictionary data files can be edited and re-written (re-formatted) using this option. The user will first be asked whether the automata or dictionary file is to be manipulated, and then whether the file is to be edited or otherwise manipulated:

Manipulate Automata or Dictionary (<cr>=Quit): DICTIONARY
Edit dictionary or Manipulate dictionary files:

or

Manipulate Automata or Dictionary (<cr>=Quit): AUTOMATA
Edit automata or Manipulate automata files:

6.1. Dictionary manipulation

6.1.1. Editing Dictionary

If editing the dictionary is chosen, then the user will be told the default lexicon (usually the Root lexicon) and given the opportunity to change it, then given a choice of editing options:

Default lexicon is Root lexicon.
Alter different lexicon? yes
Lexicon for alteration: VERBENDINGS

Add, Delete, or Edit lexicon entry? (<cr>=Quit):

Add Lexicon Entry

If ADD is chosen, a typical interaction is as follows:
Form to be added: +ed
Continuation class and gloss: # "PAST"
VERBENDINGS lexicon now contains '+ed'.

Edit Lexicon Entry

To edit a lexicon entry the user is connected to the INTERLISP editor if the option EDIT is chosen (see the section on "Dictionary file format"):

```
Entry to be edited: +ed
edit
*($("PAST"))
*1 (2 "WEAK PAST")
*p
(*"WEAK PAST")
*ok
```

Consult an INTERLISP editor manual for information on the various editing commands (* is the editor's prompt and the edit is terminated by typing "OK").

Delete Lexicon Entry

To delete such an entry, responding DELETE to the Add/Delete/Edit prompt will initiate the following kind of interaction:

```
Form to be deleted: +ed
VERBENDINGS lexicon no longer contains '+ed'.

Add, Delete, or Edit lexicon entry? (<cr>=Quit):
```

Note that after all three of these options the user is asked if one of them is to be repeated, since it would be tedious when dealing with a series of forms to return to the top level after each one. Responding with a carriage return does not take the user directly back to the top level in one fell swoop, but rather gives an opportunity to continue with other data manipulation, for example returning to the lexicon editor to edit a different lexicon, or opting to manipulate files:

```
Manipulate other data? yes
Manipulate automata or dictionary? (<cr>=Quit): DICTIONARY
Edit dictionary or Manipulate dictionary files? MANIPULATEFILES
```
6.1.2. Manipulate Dictionary Files

If the Manipulate dictionary files option is chosen, the user will be required to specify which kind of manipulation is to be performed:

Read in file, make Lisp file, or make Pascal file? (<cr>=Quit):

With all three of these file manipulating options the default filename is shown and an opportunity given to change it. Then appropriate messages are displayed when the task is complete:

Read in File [after READFILE response]

Default file is ENGLISH.DIC [in Pascal format]
Read different file? no
Dictionary file read in. [read in in Lisp format]
Quitting dictionary file manipulator...
Manipulate other data? ...

Make Lisp File [after LISPFILE response]

Default filename is ENGLISH.LISP
Make different filename? yes
Lisp dictionary filename: TESTDIC.LISP
Lisp dictionary file written.
Quitting dictionary file manipulator...
Manipulate other data? ...

Make Pascal File [after PASCALFILE response]

Default filename is ENGLISH.PAS
Make different filename? no
Pascal dictionary file written.
Quitting dictionary file manipulator...
Manipulate other data? ...

After manipulating a file the user is returned to the top level of the data manipulating option and may choose to continue with its options rather than return to the top level of KIMMO.

6.2. Automata manipulation

The automata manipulating options mirror the dictionary manipulating ones quite closely.
6.2.1. Editing Automata

If editing the automata is chosen, then a special automata editing menu is displayed:

C : Change existing automaton
M : Make new automaton
D : Delete automaton
Q : Quit

Respond with the letter associated with the option you want.

Change Existing Automaton

If C is chosen, the user will be given a menu of automaton names, each associated with a number, and asked the number of the automaton to be changed:

1 : Surface Characters
2 : Epenthesis
3 : Gemination
4 : Y-Spelling
Number of automaton to be changed?

After the appropriate number is chosen, the automaton to be changed will appear on the terminal in PASCAL format line by line. Each time a line appears on the screen, you can replace it by typing in a new line (has to be a whole line!) underneath it. Should there be no changes to that specific line, just hit the Carriage Return. A typical interaction is shown below:

[LISP :] Now you can change lines.
[LISP :] c h s S y + + = [line to be replaced]
[USER :] s h c S y + + = [new line]
[LISP :] c h s S i e 0 = [line to be replaced]
[USER :] <CR> [no changes]
[LISP :] 1 : 2 1 4 3 3 0 1 1 [line to be replaced]
[USER :] 1 : 1 1 1 1 1 1 1 1 [new line]
[LISP :] 2 : 2 3 3 3 3 0 1 1 [no changes to this line]

At the end of the automaton the user is given the ADD-LINE-MENU:

A : add new lines
N : no new lines added

Choosing A, the user can add new lines (i.e., new states) to that
automaton:

[LISP :] Now type new lines
[USER :] 7 7 7 7 7 7 7 7 [new lines to be added]
1 2 3 4 5 6 7 8
<CR>

Make New Automaton

If M is chosen, the user will first be asked to type in the name of the new automaton:

Name of New Automaton ?

After the prompt "Type row by row, terminate with empty line." just type in the new automaton in PASCAL format line by line, and terminate with an empty line:

[LISP :] Type row by row, terminate with empty line.
[USER :] a s d f g = [new automaton]
         q w e r t =
1: 1 1 1 1 1 1
2: 2 2 2 2 2 2
3: 3 3 3 3 3
<CR>

Delete-Automaton

When D is chosen, the menu of automaton names will be displayed. The user will be asked the number associated with the automaton to be deleted. After the appropriate number is given, the automaton associated with that number will be removed.

6.2.2. Manipulating Automata Files

If the automata file manipulation option is chosen, the user will be required to specify which kind of manipulation is to be performed:

Edit automata or Manipulate automata files? MANIPULATEFILES
Read in file, make Lisp file, or make Pascal file?

With these three options, as with those for manipulating the dictionary files, the user is shown the default file names and given an opportunity to change them, and appropriate messages are displayed upon completion:
Read in File  [after READFILE response]

Default file is ENGLISH.AUT
Read different file? no
Automata file read in.
Quitting automata file manipulator...
Manipulate other data? ...

Make Lisp File  [after LISPFILE response]

Default filename is ENGLISH.LISP
Make different filename? yes
Lisp automata file name: TESTAUT.LISP
Lisp automata file written.
Quitting automata file manipulator...
Manipulate other data? ...

Make Pascal File  [after PASCALFILE response]

Default filename is ENGLISH.PAS
Make different filename? no
Pascal automata file written.
Quitting automata file manipulator...
Manipulate other data? ...

Once the user has finished manipulating the data, typing "no" in response to the option to continue manipulating data will return them to the top level with a valedictory message from the data manipulator:

Manipulate other data? no
Quitting Data Manipulator...

7. N : NEW LANGUAGE

To change the language KIMMO is working with use the "N" option. This will cause the following prompt to appear:

Type in name of new language:

and when the language name is typed in the system will be reset with the filenames redefined accordingly. In addition the top-level screen will be displayed to indicate the new default filenames. Loading of the files will once again be done as and when necessary.
8. C : Change Mode

CHANGE MODE enables the user to toggle between FAST and SLOW modes (i.e., SINGLE and MULTIPLE automata modes). It requires that both the multi-automata and single-automaton files be loaded for the same language first.

The user will first receive a prompt, indicating the current mode and be asked if he/she wants to change it. When yes is chosen, if both above-mentioned files have been loaded, the user will receive another prompt indicating the change has been made. A typical interaction is as below:

Currently using multiple automata
Change? yes
Changing to a single automata

Should either one of the two files not have been loaded before, the system responds with:

Can't change yet.
Please load a single automata file first
or Please load a multi automata file first

At this point, the user should go into FILECHANGE and change the current automata file to the type of file specified. When the automata file is changed from a multi automata file to a single automaton file, and vice versa, the value of MODE changes along with it.

9. T : SET or EXAMINE TRACEPOINTS

When this option is invoked by typing "T" the system will respond with the current status of the trace options and an invitation to alter them. If they have not yet been set in the current session, the display will be as follows:

Current Trace options are:
((PROGRESS)(TIME)(LEXENTRY)(MOVEAUT)(FINALSTATE))
Change?

If the user replies "N" then they will be returned to the top-level options. Otherwise a further prompt will appear:

Tracepoint?

To which an appropriate reply would be one of the Trace option
names, whereupon the last trace prompt will appear:

Setting? (&lt;CR&gt;=OFF):

The user should type "on" to switch that trace option on, or a carriage return alone to turn it off. The user will then be returned to the top level. To confirm the alteration the Trace option can be invoked again to display the new settings.

The PROGRESS trace shows (i) the current lexicon, (ii) the state of all automata, (iii) the output so far, and (iv) remaining input characters. TIME gives the time taken for the analysis or generation, in milliseconds. In addition it displays the number of CONS operations Lisp performed. The remaining tracepoints, LEXENTRY, MOVEAUT, and FINALSTATE, provide information about various failures during the generation and analysis processes and are useful for debugging the automata and lexicon files. LEXENTRY displays the dictionary entry the system is currently looking at and trying to match with the input. A MOVEAUT trace is printed whenever an automaton blocks. It shows the name of the automaton, state, and the pair of input characters rejected. FINALSTATE indicates the automaton and a state that prevents a form from being accepted because it is not final.