ACL2(ml): Machine Learning for ACL2

J. Heras, E. Komendantskaya, E. Maclean, and M. Johansson.
Queries to jonathanheras@computing.dundee.ac.uk

August 5, 2013

Abstract

This manual describes ACL2(ml), a machine-learning extension for ACL2, a description of this tool and its most relevant features can be found in [1].

Contents

1 Requirements 1

2 Installation 2

3 Using ACL2(ml) 2
- 3.1 Getting started 2
- 3.2 Algorithms menu 4
- 3.3 Granularity menu 4
- 3.4 Export Library 4
- 3.5 Available libraries for clustering 4
- 3.6 Show similar theorems 5
- 3.7 Clusters 5
- 3.8 Additional features 7
  - 3.8.1 Activate lemma analogy tool 7
  - 3.8.2 Explain cluster similarities 7

1 Requirements

Before installing ACL2(ml), you need to download, install and configure the following software.

- ACL2.
- Emacs.
- OCaml (it is optional but necessary for the automatic discovery of auxiliary lemmas, see Section 3.8.1).

You can follow the instructions presented in `http://www.cs.utexas.edu/users/moore/publications/acl2-programming-exercises1.html`.

2 Installation

Before using ACL2(ml), it is necessary to configure some variables. First of all, open your *.emacs* file. This file is usually located in */home/user/.emacs.* At the end of the file include the line:

```
(load-file "ACL2(ml)-location/main.el")
```

where `ACL2(ml)-location` must be changed with the path to the folder where you have downloaded ACL2(ml).

Now, go to the folder where you have downloaded ACL2(ml) and open the file *main.el.* Modify the constant **home-dir**

- *home-dir*: you must replace the current path assigned to this constant with the path where you have downloaded ACL2(ml).

Now, if you want to use the lemma analogy tool, see Section 3.8.1, you need to open a command line and go to the folder ACL2(ml)/lemmaanalogy (where ACL2(ml) is the folder where you have downloaded ACL2(ml)). Then, use `make` to compile the lemma analogy tool.

This finishes the installation of ACL2(ml).

3 Using ACL2(ml)

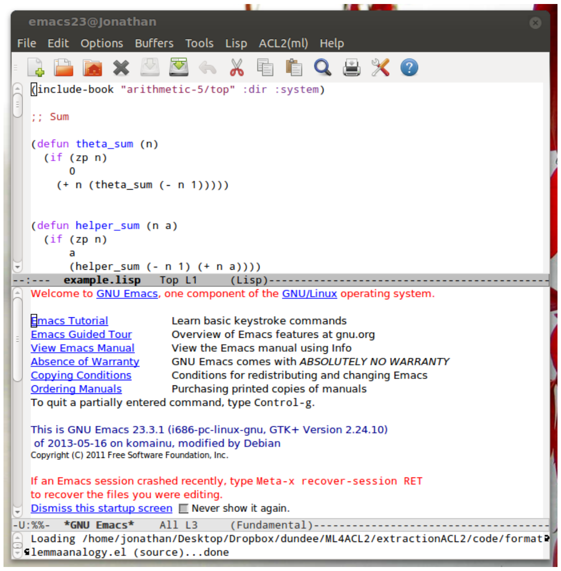
To illustrate the use of ACL2(ml), we will use the file `example.lisp` which can be find in the same folder of this manual. The example `example.lisp` includes the definition of several recursive and tail recursive arithmetic functions and the proof of their equivalences.

3.1 Getting started

Open file `example.lisp` using Emacs and run ACL2 (M-x run-acl2) If you have installed everything properly, the image of Figure 1 will appear. The only difference with the usual emacs interface is a menu called *ACL2(ml).*

This menu contains several options to configure ACL2(ml). First of all, you can activate the buttons and the different libraries which can be used pressing the options: *Activate icons* and *Show Available libraries.* Once that you have done this, the interface looks like in Figure 2.

Let us explain the functionality of the options included in the ACl2(ml) menu.



```
(include-book "arithmetic-5/top" :dir :system)

;; Sum

(defun theta_sum (n)
  (if (zp n)
      0
    (+ n (theta_sum (- n 1)))))


(defun helper_sum (n a)
  (if (zp n)
      a
      (helper_sum (- n 1) (+ n a))))
--:---  example.lisp   Top L1     (Lisp)----------------------------------------
```

Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial	Learn basic keystroke commands
Emacs Guided Tour	Overview of Emacs features at gnu.org
View Emacs Manual	View the Emacs manual using Info
Absence of Warranty	GNU Emacs comes with *ABSOLUTELY NO WARRANTY*
Copying Conditions	Conditions for redistributing and changing Emacs
Ordering Manuals	Purchasing printed copies of manuals

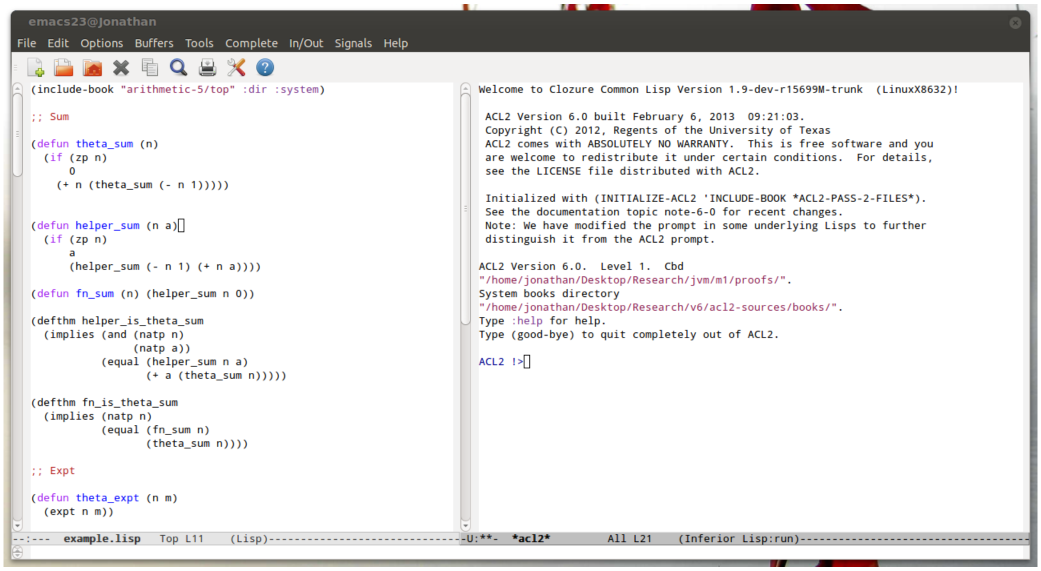
To quit a partially entered command, type `Control-g`.

This is GNU Emacs 23.3.1 (i686-pc-linux-gnu, GTK+ Version 2.24.10)
of 2013-05-16 on komainu, modified by Debian
Copyright (C) 2011 Free Software Foundation, Inc.

If an Emacs session crashed recently, type `Meta-x recover-session RET`
to recover the files you were editing.
Dismiss this startup screen ☐ Never show it again.

```
-U:%%-  *GNU Emacs*    All L3     (Fundamental)----------------------------------
Loading /home/jonathan/Desktop/Dropbox/dundee/ML4ACL2/extractionACL2/code/format
lemmaanalogy.el (source)...done
```

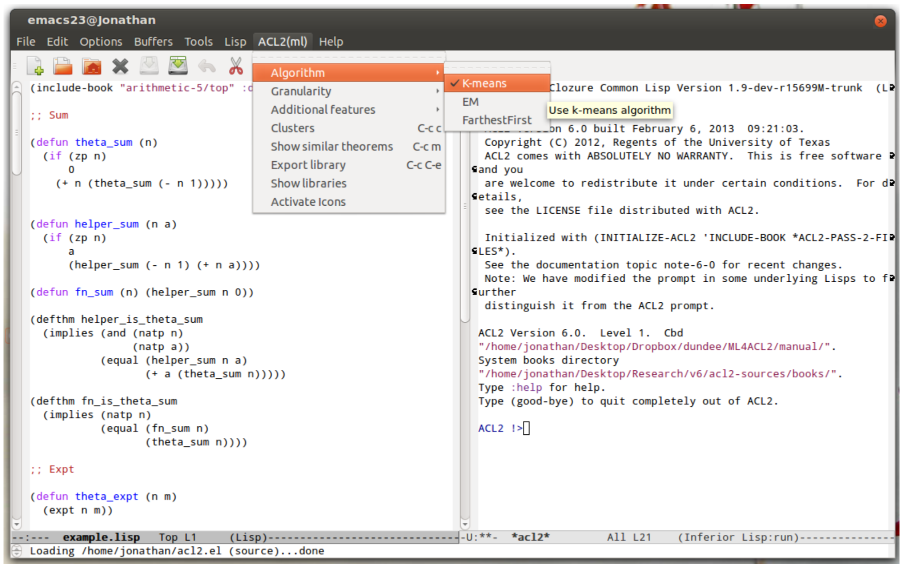
Figure 1: Initial screen of ACL2(ml).


emacs23@Jonathan
File Edit Options Buffers Tools Complete In/Out Signals Help

```
(include-book "arithmetic-5/top" :dir :system)

;; Sum

(defun theta_sum (n)
  (if (zp n)
      0
    (+ n (theta_sum (- n 1)))))


(defun helper_sum (n a)
  (if (zp n)
      a
      (helper_sum (- n 1) (+ n a))))

(defun fn_sum (n) (helper_sum n 0))

(defthm helper_is_theta_sum
  (implies (and (natp n)
                (natp a))
           (equal (helper_sum n a)
                  (+ a (theta_sum n)))))

(defthm fn_is_theta_sum
  (implies (natp n)
           (equal (fn_sum n)
                  (theta_sum n))))

;; Expt

(defun theta_expt (n m)
  (expt n m))
--:---  example.lisp   Top L11    (Lisp)-----------------------------
```

```
Welcome to Clozure Common Lisp Version 1.9-dev-r15699M-trunk  (LinuxX8632)!

 ACL2 Version 6.0 built February 6, 2013  09:21:03.
 Copyright (C) 2012, Regents of the University of Texas
 ACL2 comes with ABSOLUTELY NO WARRANTY.  This is free software and you
 are welcome to redistribute it under certain conditions.  For details,
 see the LICENSE file distributed with ACL2.

 Initialized with (INITIALIZE-ACL2 'INCLUDE-BOOK *ACL2-PASS-2-FILES*).
 See the documentation topic note-6-0 for recent changes.
 Note: We have modified the prompt in some underlying Lisps to further
 distinguish it from the ACL2 prompt.

ACL2 Version 6.0.  Level 1.  Cbd
"/home/jonathan/Desktop/Research/jvm/m1/proofs/".
System books directory
"/home/jonathan/Desktop/Research/v6/acl2-sources/books/".
Type :help for help.
Type (good-bye) to quit completely out of ACL2.

ACL2 !>
-U:**-  *acl2*        All L21    (Inferior Lisp:run)-----------------------------
```

Figure 2: ACL2(ml) interface with all the options activated.



```lisp
(include-book "arithmetic-5/top" :d

;; Sum

(defun theta_sum (n)
  (if (zp n)
      0
    (+ n (theta_sum (- n 1)))))


(defun helper_sum (n a)
  (if (zp n)
      a
      (helper_sum (- n 1) (+ n a))))

(defun fn_sum (n) (helper_sum n 0))

(defthm helper_is_theta_sum
  (implies (and (natp n)
                (natp a))
           (equal (helper_sum n a)
                  (+ a (theta_sum n)))))

(defthm fn_is_theta_sum
  (implies (natp n)
           (equal (fn_sum n)
                  (theta_sum n))))

;; Expt

(defun theta_expt (n m)
  (expt n m))
```

```
--:---  example.lisp   Top L1     (Lisp)----------------------------
Loading /home/jonathan/acl2.el (source)...done
```

```
Clozure Common Lisp Version 1.9-dev-r15699M-trunk  (L
on 6.0 built February 6, 2013  09:21:03.
 Copyright (C) 2012, Regents of the University of Texas
 ACL2 comes with ABSOLUTELY NO WARRANTY.  This is free software 
and you
 are welcome to redistribute it under certain conditions.  For d
etails,
 see the LICENSE file distributed with ACL2.

 Initialized with (INITIALIZE-ACL2 'INCLUDE-BOOK *ACL2-PASS-2-FI
LES*).
 See the documentation topic note-6-0 for recent changes.
 Note: We have modified the prompt in some underlying Lisps to f
urther
 distinguish it from the ACL2 prompt.

ACL2 Version 6.0.  Level 1.  Cbd
"/home/jonathan/Desktop/Dropbox/dundee/ML4ACL2/manual/".
System books directory
"/home/jonathan/Desktop/Research/v6/acl2-sources/books/".
Type :help for help.
Type (good-bye) to quit completely out of ACL2.

ACL2 !>
```

```
-U:**-  *acl2*       All L21    (Inferior Lisp:run)--------------
```

Figure 3: The Algorithms submenu.

3.2 Algorithms menu

The user can select different algorithms to obtain similar lemmas. The algorithms which are available are: K-means, EM and FarthestFirst, see Figure 3.

3.3 Granularity menu

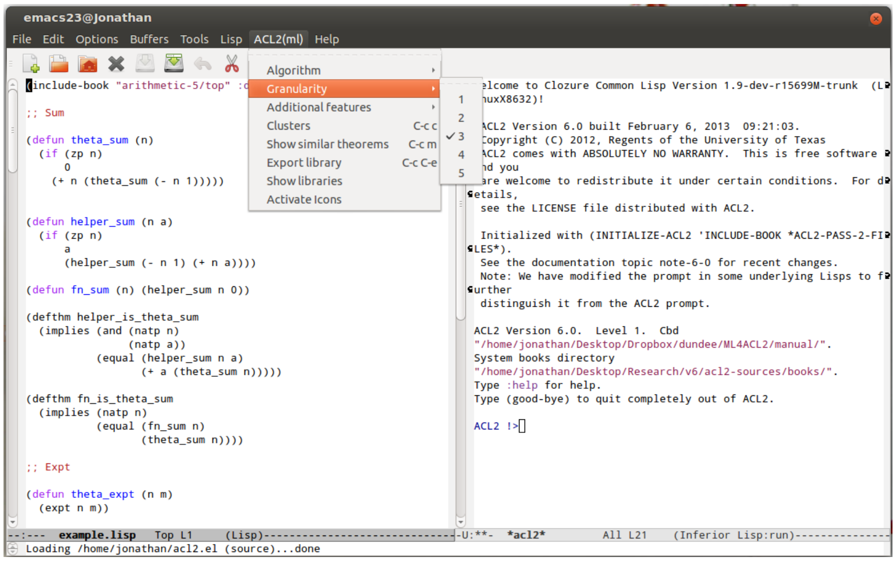
This option allows the user to select the granularity of the families of similar lemmas, by selecting a value between 1 and 5, where 1 stands for a low granularity (producing big and general families of similar lemmas) and 5 stands for a high granularity (producing small and precise families of similar lemmas). See Figure 4. The granularity parameter was first described for ML4PG, a machine-learning extension for Coq. A detailed description of ML4PG and the granularity parameter can be read in [2].

3.4 Export Library

Using the Export library option, the user can export the library for further use.

3.5 Available libraries for clustering

This option allows the user to find families of similar lemmas across several libraries previously exported.



```
(include-book "arithmetic-5/top" :d

;; Sum

(defun theta_sum (n)
  (if (zp n)
      0
    (+ n (theta_sum (- n 1)))))


(defun helper_sum (n a)
  (if (zp n)
      a
      (helper_sum (- n 1) (+ n a))))

(defun fn_sum (n) (helper_sum n 0))

(defthm helper_is_theta_sum
  (implies (and (natp n)
                (natp a))
           (equal (helper_sum n a)
                  (+ a (theta_sum n)))))

(defthm fn_is_theta_sum
  (implies (natp n)
           (equal (fn_sum n)
                  (theta_sum n))))

;; Expt

(defun theta_expt (n m)
  (expt n m))
--:---  example.lisp    Top L1     (Lisp)------------------------------
Loading /home/jonathan/acl2.el (source)...done
```

```
elcome to Clozure Common Lisp Version 1.9-dev-r15699M-trunk  (L
nuxX8632)!

 ACL2 Version 6.0 built February 6, 2013  09:21:03.
 Copyright (C) 2012, Regents of the University of Texas
 ACL2 comes with ABSOLUTELY NO WARRANTY.  This is free software
 nd you
 are welcome to redistribute it under certain conditions.  For d
etails,
 see the LICENSE file distributed with ACL2.

 Initialized with (INITIALIZE-ACL2 'INCLUDE-BOOK *ACL2-PASS-2-FI
LES*).
 See the documentation topic note-6-0 for recent changes.
 Note: We have modified the prompt in some underlying Lisps to f
urther
 distinguish it from the ACL2 prompt.

ACL2 Version 6.0.  Level 1.  Cbd
"/home/jonathan/Desktop/Dropbox/dundee/ML4ACL2/manual/".
System books directory
"/home/jonathan/Desktop/Research/v6/acl2-sources/books/".
Type :help for help.
Type (good-bye) to quit completely out of ACL2.

ACL2 !>
-U:**-  *acl2*       All L21     (Inferior Lisp:run)--------------
```

Figure 4: ACL2(ml) granularity menu.

3.6 Show similar theorems

We can cluster libraries relative to a concrete proof. An example using the `example.lisp` library with the options:

- Algorithm: K-means,
- Granularity: 3,

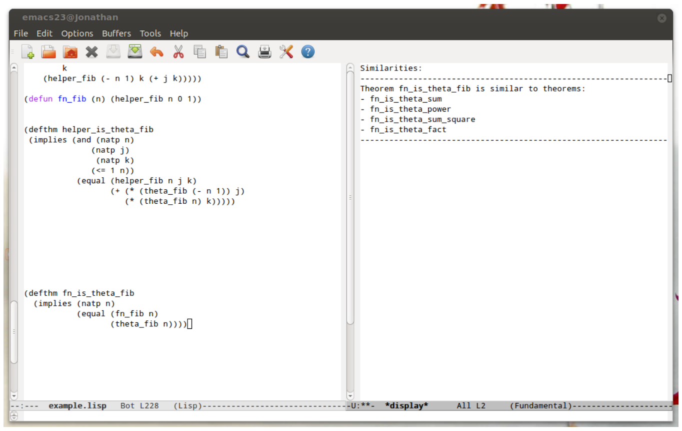
and with the lemma `fn-is-theta-fib` (at the end of the file) is shown in Figure 5. Put the cursor at the end of lemma `fn-is-theta-fib` and press the second most right button (or select the option Show Similar Theorems in the ACL2(ml) menu).

This functionality can also invoked using the right most button of the Proof General toolbar.

3.7 Clusters

The option *Show Clusters* of the Statistics menu shows families of similar lemmas when a library is clustered irrespective of the current proof goal. An example using the `example.lisp` library with the options:

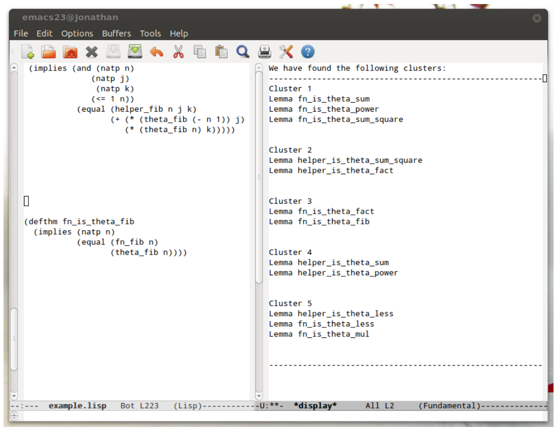
- Algorithm: K-means,
- Granularity: 5,


emacs23@Jonathan
File Edit Options Buffers Tools Help
--:--- example.lisp   Bot L228   (Lisp)
-U:**- *display*   All L2   (Fundamental)

```
        k
    (helper_fib (- n 1) k (+ j k)))))

(defun fn_fib (n) (helper_fib n 0 1))


(defthm helper_is_theta_fib
 (implies (and (natp n)
              (natp j)
               (natp k)
              (<= 1 n))
           (equal (helper_fib n j k)
                  (+ (* (theta_fib (- n 1)) j)
                     (* (theta_fib n) k)))))







(defthm fn_is_theta_fib
  (implies (natp n)
           (equal (fn_fib n)
                  (theta_fib n))))
```

```
Similarities:
-------------------------------------------------------------------
Theorem fn_is_theta_fib is similar to theorems:
- fn_is_theta_sum
- fn_is_theta_power
- fn_is_theta_sum_square
- fn_is_theta_fact
-------------------------------------------------------------------
```

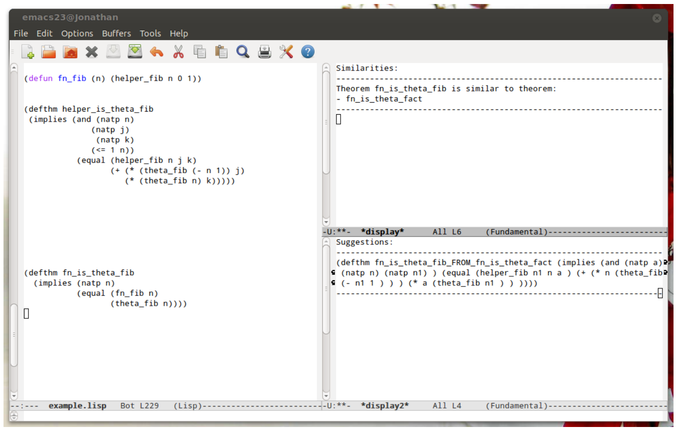
Figure 5: On the right side, several suggestions provided by ACL2(ml) related to lemma `fn-is-theta-fib`. The Proof General window has been split into two windows positioned side by side: the left one keeps the current script, and the right one shows the families of similar lemmas to `fn-is-theta-fib`.


emacs23@Jonathan
File Edit Options Buffers Tools Help
--:--- example.lisp   Bot L223   (Lisp)
-U:**- *display*   All L2   (Fundamental)

```
 (implies (and (natp n)
              (natp j)
               (natp k)
              (<= 1 n))
           (equal (helper_fib n j k)
                  (+ (* (theta_fib (- n 1)) j)
                     (* (theta_fib n) k)))))




(defthm fn_is_theta_fib
  (implies (natp n)
           (equal (fn_fib n)
                  (theta_fib n))))
```

```
We have found the following clusters:
-------------------------------------------------------------------
Cluster 1
Lemma fn_is_theta_sum
Lemma fn_is_theta_power
Lemma fn_is_theta_sum_square


Cluster 2
Lemma helper_is_theta_sum_square
Lemma helper_is_theta_fact


Cluster 3
Lemma fn_is_theta_fact
Lemma fn_is_theta_fib


Cluster 4
Lemma helper_is_theta_sum
Lemma helper_is_theta_power


Cluster 5
Lemma helper_is_theta_less
Lemma fn_is_theta_less
Lemma fn_is_theta_mul


-------------------------------------------------------------------
```

Figure 6: Clusters for the example.lisp library. The Proof General window has been split into two windows positioned side by side: the left one keeps the current script, and the right one shows the families of similar lemmas.



```
(defun fn_fib (n) (helper_fib n 0 1))


(defthm helper_is_theta_fib
 (implies (and (natp n)
              (natp j)
               (natp k)
              (<= 1 n))
          (equal (helper_fib n j k)
                 (+ (* (theta_fib (- n 1)) j)
                    (* (theta_fib n) k)))))


(defthm fn_is_theta_fib
  (implies (natp n)
           (equal (fn_fib n)
                  (theta_fib n))))
```

```
--:---  example.lisp   Bot L229   (Lisp)------------------------
```

```
Similarities:
--------------------------------------------------------------------
Theorem fn_is_theta_fib is similar to theorem:
- fn_is_theta_fact
--------------------------------------------------------------------
```

```
-U:**-  *display*     All L6     (Fundamental)-----------------------
```

```
Suggestions:
--------------------------------------------------------------------
(defthm fn_is_theta_fib_FROM_fn_is_theta_fact (implies (and (natp a)
 (natp n) (natp n1) ) (equal (helper_fib n1 n a ) (+ (* n (theta_fib
 (- n1 1 ) ) ) (* a (theta_fib n1 ) ) ))))
--------------------------------------------------------------------
```

```
-U:**-  *display2*    All L4     (Fundamental)-----------------------
```

Figure 7: On the bottom right side, suggestion to prove lemma `fn-is-theta-fib`.

is shown in Figure 6.

This functionality can also invoked using the second right most button of the Emacs toolbar.

3.8 Additional features

3.8.1 Activate lemma analogy tool

If this experimental feature is activated, ACL2(ml) is able to show suggestions to prove the current lemma on the basis of the proofs of other similar lemmas. See Figure 7 (parameters: k-means algorithm, 5 as granularity value).

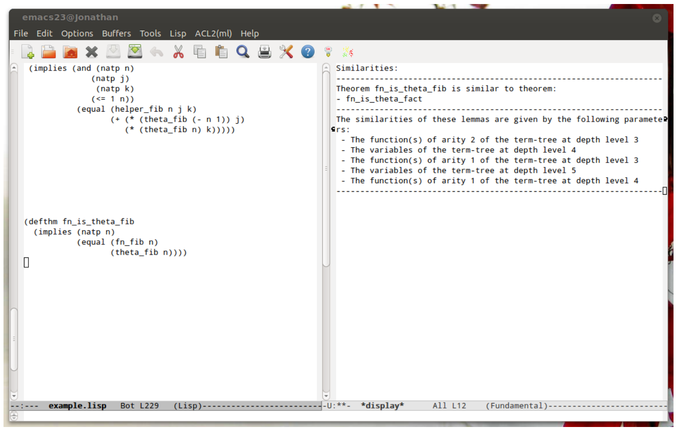
3.8.2 Explain cluster similarities

If this option is activated, ACL2(ml) shows the reason because the different lemmas are grouped in the same cluster, see Figure 8.

References

[1] J. Heras, M. Johansson, E. Komendantskaya, and E. Maclean. Proof-pattern recognition in acl2. 2013. `http://staff.computing.dundee.ac.uk/jheras/acl2ml/`.

[2] E. Komendantskaya, J. Heras, and G. Grov. Machine learning in proof general: interfacing interfaces. *Electronic Proceedings in Theoretical Computer Science.*



```
 (implies (and (natp n)
               (natp j)
                (natp k)
               (<= 1 n))
          (equal (helper_fib n j k)
                 (+ (* (theta_fib (- n 1)) j)
                    (* (theta_fib n) k)))))




(defthm fn_is_theta_fib
  (implies (natp n)
           (equal (fn_fib n)
                  (theta_fib n))))
```

```
Similarities:
------------------------------------------------------------------
Theorem fn_is_theta_fib is similar to theorem:
- fn_is_theta_fact
------------------------------------------------------------------
The similarities of these lemmas are given by the following paramete
rs:
 - The function(s) of arity 2 of the term-tree at depth level 3
 - The variables of the term-tree at depth level 4
 - The function(s) of arity 1 of the term-tree at depth level 3
 - The variables of the term-tree at depth level 5
 - The function(s) of arity 1 of the term-tree at depth level 4
------------------------------------------------------------------
```

--:--- example.lisp   Bot L229   (Lisp)-------------------------
-U:**- *display*   All L12   (Fundamental)-----------------------

Figure 8: On the right side, several suggestions provided by ACL2(ml) related to lemma `fn-is-theta-fib` and the parameters for the similarity.