

Rfuzzy (and Ciao Prolog) installation manual

Víctor Pablos Ceruelo

e-mail: victorpablosceruelo at gmail.com

web page: <http://babel.ls.fi.upm.es/~vpablos>

October 13, 2010

Contents

1	About Rfuzzy	3
2	Install Ciao Prolog	3
2.1	Easy Installation (Debian and Ubuntu)	3
2.2	Not so easy installation	3
2.2.1	Install cowbuilder	3
2.2.2	Add to pbuilderrc the mirrors you have in sources.list . . .	3
2.2.3	Download ciao-prolog sources	3
2.2.4	Initialize cowbuilder environment, update it and build . .	4
2.2.5	Results from compilation	4
3	Installing the rfuzzy library	4
4	Examples	4
5	Rfuzzy beginning	5

1 About Rfuzzy

Rfuzzy is a preprocessor package hosted at <http://babel.ls.fi.upm.es/software/rfuzzy/> that has been tested for Ciao Prolog 1.13+r11293.

More information can be found there, although for semantics we recommend you the following papers: [MHPCS10] [VÁS10] [ÁSV10] [SMHP09] [MHPS09] [PSM09] [PMS08]

2 Install Ciao Prolog

2.1 Easy Installation (Debian and Ubuntu)

Take a look at <https://babel.ls.fi.upm.es/~vpablos/others/packages/readme.html>. Since this version of Rfuzzy has been tested for Ciao Prolog 1.13+r11293, we recommend you to install this and only this version of Ciao Prolog.

2.2 Not so easy installation

2.2.1 Install cowbuilder

Use apt-get, aptitude or synaptic to install cowbuilder.

2.2.2 Add to pbuilder the mirrors you have in sources.list

Simple edit '/etc/pbuilderrc' and add the mirrors you have enabled at your '/etc/apt/sources.list' file in the 'OTHERMIRROR' option. We have used the following lines to build ubuntu gutsy i386 precompiled packages:

/etc/pbuilderrc OTHERMIRROR option for ubuntu Gutsy

```
OTHERMIRROR="deb http://archive.ubuntu.com/ubuntu/ gutsy restricted
universe multiverse | deb-src http://archive.ubuntu.com/ubuntu/ gutsy re-
stricted universe multiverse | deb http://archive.canonical.com/ubuntu gutsy
partner | deb-src http://archive.canonical.com/ubuntu gutsy partner | deb
http://security.ubuntu.com/ubuntu/ gutsy-security main restricted universe
multiverse | deb-src http://security.ubuntu.com/ubuntu/ gutsy-security main
restricted universe multiverse | deb http://archive.ubuntu.com/ubuntu/
gutsy-updates main restricted universe multiverse | deb-src
http://archive.ubuntu.com/ubuntu/ gutsy-updates main restricted uni-
verse multiverse"
```

2.2.3 Download ciao-prolog sources

Download ciao-prolog sources from <https://babel.ls.fi.upm.es/~vpablos/others/packages/sources/>. The files you need are listed below:

Ciao prolog sources' files

```
ciao-prolog_1.13+svn20080125-1.diff.gz ... 10.4K
ciao-prolog_1.13+svn20080125-1.dsc ... 0.6K
ciao-prolog_1.13+svn20080125.orig.tar.gz ... 30.9M
```

2.2.4 Initialize cowbuilder environment, update it and build

Initialize cowbuilder environment and update it to the last version. After that you can do your precompiled packages with the option `-build`:

Cowbuilder commands to make precompiled files

```
cowbuilder -create
cowbuilder -update
cowbuilder -build ciao-prolog_1.13+svn20080125-1.dsc
```

2.2.5 Results from compilation

Finally you'll obtain at `/var/cache/pbuilder/result/` all the files you need to install ciao prolog. Use `dpkg` or other tool to install them and you're done.

3 Installing the rfuzzy library

Uncompress the `.tgz` file and copy the `rfuzzy` subfolder into the ciao library installation folder. It is usually `/usr/lib/ciao/ciao-1.13/library/` or `/usr/local/lib/ciao/ciao-1.13/library/`.

Simply drag and drop the `rfuzzy` subfolder there. That's all. Be sure all users can read those files and folder (use `chmod 644 filename` for the files and `chmod 755 filename` for the folder).

For debugging purposes we recommend you to install too the debugging package, which is included in the `.tgz` file. For that simply copy the `debugger_pkg` subfolder into the ciao library installation folder.

4 Examples

At subfolder `examples` you have some examples that show you how to use the library. They are:

Examples included

```
good_player.pl
human_development.pl
jobs.pl
restaurant.pl
teams.pl
which_row.pl
```

5 Rfuzzy beginning

To start using the `rfuzzy` package you need to include the following line in your program source file:

Rfuzzy package beginning

```
:- module(file_name,_,[rfuzzy, clpr]).
```

If you want to see what is being executed by the Ciao compiler, the easiest way is to install the *debugger_pkg* and include the following line (instead of the previous one).

Rfuzzy package beginning

```
:- module(file_name,_,[rfuzzy, clpr, debugger_pkg]).
```

The following example shows how this is done:

```
:- module(good_player,_,[rfuzzy, clpr, debugger_pkg]).

% Define the individuals belonging to the set player.
player(john).
player(karl).
player(mike).
player(lebron).
player(deron).
player(damian).
player(aito).

% Define that only player are valid individuals for
% the fuzzy set good_player.
:- set_prop good_player/1 => player/1.

% An individual is a good player with a truth value of 0.1
% if we can not compute a more accurate value.
:- default(good_player/1, 0.1).

% The rule to determine the grade of belonging of
% a player to the fuzzy set of good_player has a
% confidence of 0.8. Its result is a combination of
% how much (the truth value) he/she is swift, tall and
% an experienced player.
good_player(J) cred (prod,0.8) :~ prod((swift(J), tall(J), experience(J))).

:- set_prop experience/1 => player/1.
:- default(experience/1, 0.1).

experience(lebron) value 0.4.
experience(deron) value 0.3.

% Tall predicate depending on the player's height.
tall(Player, V) :- height(Player, H), tall_func(H, V).
tall_func :# ([ (1500, 0), (1800, 0.5), (2000, 1), (3000, 1) ]).
```

```

% Height of players in mm.
height(john, 1700).
height(marcus, 1780).
height(aito, 1800).
height(damian, 1850).
height(karl, 1900).
height(lebron, 1950).
height(deron, 2000).

:- set_prop swift/1 => player/1.
:- default(swift/1,0.5).

swift(john) value 1.
swift(karl) value 0.6 .
swift(mike) value 0.9 .
swift(lebron) value 1.
swift(deron) value 0.8.

% Queries (examples).

% ?- good_player(X, Y).
% ?- good_player(X, rat(4,125)).

```

Enjoy !!!

References

- [ÁSV10] Álvaro Fernández-Díaz, Susana Muñoz-Hernández, and Víctor Pablos-Ceruelo. Developing user-sensitive e-commerce applications from fuzzy concepts. In *Algebraic methods, soft computing, and program verification*, International Center for Mathematical Meetings (CIEM), Castro Urdiales, Cantabria, Spain, April 2010. Julio Rubio (Univ. de La Rioja) and Manuel Ojeda-Aciego (Univ. de Málaga). Sponsored by: Ingenio2010, CIEM, Excmo. Ayuntamiento de Castro Urdiales. Keynote Speakers: Thomas Arts (Univ. of Gothenburg), Gilles Barthe (Madrid Institute for Advanced Studies (Software)), Stefan Berghofer (Technische Universität München), Mario de Jesús Pérez Jiménez (Univ. de Sevilla), José Luis Ruiz Reina (Univ. de Sevilla), Enric Trillas (European Centre for Soft Computing). 3
- [MHPCS10] Susana Munoz-Hernandez, Víctor Pablos-Ceruelo, and Hannes Strass. Rfuzzy: Syntax, semantics and implementation details of a simple and expressive fuzzy tool over prolog. *Information Sciences*, In Press, Accepted Manuscript:–, 2010. 3
- [MHPS09] Susana Muñoz-Hernández, Víctor Pablos-Ceruelo, and Hannes Strass. Rfuzzy: An expressive simple fuzzy compiler. In Joan

- Cabestany, Francisco Sandoval, Alberto Prieto, and Juan M. Corchado, editors, *IWANN (1)*, volume 5517 of *Lecture Notes in Computer Science*, pages 270–277. Springer, 2009. 3
- [PMS08] Víctor Pablos-Ceruelo, Susana Muñoz Hernandez, and Hannes Strass. Rfuzzy framework. In Puri Arenas and Damiano Zanardini, editors, *18th Workshop on Logic-based methods in Programming Environments, WLPE 2008*, pages 62–76, Udine, Italy, December 2008. 3
- [PSM09] Víctor Pablos-Ceruelo, Hannes Strass, and Susana Muñoz Hernández. Rfuzzy—a framework for multi-adjoint fuzzy logic programming. In *Fuzzy Information Processing Society, 2009. NAFIPS 2009. Annual Meeting of the North American Fuzzy Information Processing Society Annual Conference*, pages 1–6, Cincinnati, Ohio, USA, June 2009. 3
- [SMHP09] Hannes Strass, Susana Muñoz-Hernández, and Víctor Pablos-Ceruelo. Operational semantics for a fuzzy logic programming system with defaults and constructive answers. In João Paulo Carvalho, Didier Dubois, Uzay Kaymak, and João Miguel da Costa Sousa, editors, *IFSA/EUSFLAT Conf.*, pages 1827–1832, 2009. 3
- [VÁS10] Víctor Pablos-Ceruelo, Álvaro Fernández-Díaz, and Susana Muñoz-Hernández. Developing user-sensitive search engines from fuzzy concepts. In *JISBD'2010*, September 2010. accepted for publication, see <http://www.dsic.upv.es/conferences/jisbd2010/articulos.shtml>. 3