

Vaucanson, a generic library for finite state machines. Copyright (C) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2011 The Vaucanson Group.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

The complete GNU General Public Licence Notice can be found as the 'COPYING' file in the root directory.

The Vaucanson Group consists of people listed in the 'AUTHORS' file.

Introduction to Vaucanson

[Vaucanson](#), a C++ generic library for weighted finite state machines.

Contents

Introduction to Vaucanson	1
Overview	1
Installation	1
Requirements	2
Libraries installed in non-standard directories	2
Graph implementations	3
Layout of the tarball	3
Using Vaucanson	3
See Also	4
Licence	4
Contacts	4

Overview

Initiated by Jacques Sakarovitch in 2000, Vaucanson is a project developed by the [École Nationale Supérieure des Télécommunications](#) and the [EPITA Research and Development Laboratory \(LRDE\)](#).

The goal of this library is to enable the development of C++ programs manipulating weighted finite automata in an abstract and general way with, at the same time, a large specialization power. On the one hand, we can write algorithms working on every automaton with weights in any semiring and with words from any free monoids. And on the other hand, a particular algorithm can be specialized for a particular data structure.

Yet, Vaucanson is an ongoing development project. Therefore algorithms, data structures and the general architecture are not totally stable and well tested.

Please send any question or comments to vaucanson@lrde.epita.fr, and report bugs to either our issue tracker <http://vaucanson.lrde.org/>, or to vaucanson-bugs@lrde.epita.fr.

Installation

To install Vaucanson on your system, type in the classical sequence at the command prompt:

```
./configure
make
make install (as root)
```

Note that an installation is specific to the compiler used to install it. Indeed, the call to `./configure` enables some workarounds and, consequently, users must compile with the same compiler to avoid compatibility problems.

Between `make` and `make install`, you may also want to run:

```
make demos
make sanity-check
make check
```

`make demos` will build example binaries in `src/demos/`. `make sanity-check` will make sure that Vaucanson's header files can be included and compiled. `make check` will run the test suite to check the whole library. Running the test suite may require up to 10GB of free space and several hours.

Requirements

Vaucanson was tested with the [GNU Compiler Collection \(GCC\)](#) version 4.1.x to 4.6.x.

TAF-Kit and some test cases can use the AT&T dot format to save automaton in a human readable file. You should install [Graphviz](#) to visualize these `.dot` files or run the test suite.

The XML I/O system is based on the Apache [Xerces-C++](#) library version 2.7 or above.

The C++ Application Binary Interface (ABI) of the Xerces-C++ library must be the same as the C++ ABI of the compiler used to build Vaucanson's XML I/O system. In particular, users of Fink or DarwinPorts on MacOS should pay attention to the compiler that was used to build their version of the Xerces-C++ library, as it might differ from the one used to build Vaucanson. Vaucanson should work with any version after 2.3.

[Boost](#) has been used since Vaucanson 1.1. It is a C++ library which provides many useful objects, including hash tables. Currently, Boost is used in algorithms only, but its use shall be extended to automata structures and other portions of code. You must install this library on your system. Vaucanson should support any version after 1.34.

[Ncurses](#) is used by the interactive editor of TAF-Kit. We use only a very small part of its stable interface so any version should work. If [Ncurses](#) is not installed on your system, the compilation of TAF-Kit will be disabled.

Libraries installed in non-standard directories

If you have installed Xerces-C++ or Boost in a non-standard directory (i.e., a directory that is not searched by default by your C++ compiler), you will have to set the `CPPFLAGS` and `LDFLAGS` variables to pass the necessary `-I` and `-L` options to the preprocessor and linker.

For instance if you installed Xerces-C++ in `/opt/xerces/` and Boost in `/opt/boost/` you should run `./configure` as follows:

```
./configure CPPFLAGS="-I/opt/xerces -I/opt/boost" LDFLAGS="-L/opt/xerces -L/opt/boost"
```

Graph implementations

Vaucanson can use two graph implementations: `listg` is a representation based on adjacency lists, while `bmig` is a representation using Boost Multi-Index containers. The default implementation is `listg` and you can select the other with:

```
./configure --default-graph-impl=bmig
```

For further configure options, type:

```
./configure --help
```

Layout of the tarball

The Vaucanson project directory layout is as follows:

build-aux Auxiliary tools used by the GNU Build System during `configure` and `make` stages.

data Data files to be installed on your system. They include an XML schema, example automata, and Emacs customizations.

debian Data to generate Debian packages.

doc The documentation.

manual The Taf-Kit manual.

ref [Doxygen](#) documentation (automatically generated from the source code)

gnulib Portability functions from the `gnulib` library.

lib Instantiation of some contexts as libraries.

include The code of the Vaucanson C++ Library.

m4 Portability macros from the `gnulib` library.

src Benchmarks, demonstration executables, and test cases.

tools Developer scripts.

taf-kit TAF-Kit sources and tests.

cbs C++ Benchmarking Suite.

Using Vaucanson

Vaucanson comes with several demos. Looking at them is a good way to see what Vaucanson can do and how it works. They can be found in the `src/demos` directory.

The TAF-Kit (Typed Automata Functions) documentation can be found in `doc/manual`.

See Also

There are other sources of interest in the distribution.

- Headline news about the project can be found in the file `NEWS` at the root of the source tree.
- The library reference manual, generated by [Doxygen](#), is located in `doc/ref`.
- Information about the test suite generation mechanism can be found in the file `src/tests/test-suites/README`.

Licence

Vaucanson is now released under the GNU General Public Licence. See the file `COPYING` (at the root of the source tree) for details.

Vaucanson was released under the GNU Lesser General Public Licence until version 0.7.

Contacts

The team can be reached by mail at vaucanson@lrde.epita.fr. Snail mail addresses follow.

- Vaucanson - ENST

Jacques Sakarovitch
Ecole nationale supérieure des télécommunications
46, rue Barrault
75634 Paris CEDEX 13
France

- Vaucanson - IGR

Sylvain Lombardy
Laboratoire d'informatique de l'Institut Gaspard Monge
Université Paris-Est Marne-la-Vallée
77454 Marne-la-Vallée CEDEX 2
France

- Vaucanson - LRDE

Laboratoire de Recherche et Développement de l'EPITA
14-16 rue Voltaire
94276 Le Kremlin-Bicêtre CEDEX
France