

[Skip to main content](#)

[__Back to top](#)

[__`Ctrl`+`K`](#)

[[!\[conda 25.1.2.dev29 documentation -](#)

[Home\]\(../../../../../_static/conda_logo_full.svg\) \]\(../../../../../index.html\)](#)

* [\[Conda\]\(https://docs.conda.io/projects/conda/\)](#)

* [\[Conda-build\]\(https://docs.conda.io/projects/conda-build\)](#)

* [\[Miniconda\]\(https://docs.anaconda.com/free/miniconda/\)](#)

* [\[conda.org\]\(https://conda.org\)](#)

* [\[__ GitHub\]\(https://github.com/conda/conda "GitHub"\)](#)

* [\[!\[Element\]\(../../../../../_static/element_logo.svg\)\]\(http://bit.ly/conda-chat-room "Element"\)](#)

* [\[__ Discourse\]\(https://conda.discourse.group/ "Discourse"\)](#)

* [\[Conda\]\(https://docs.conda.io/projects/conda/\)](#)

* [\[Conda-build\]\(https://docs.conda.io/projects/conda-build\)](#)

* [\[Miniconda\]\(https://docs.anaconda.com/free/miniconda/\)](#)

* [\[conda.org\]\(https://conda.org\)](#)

* [\[__ GitHub\]\(https://github.com/conda/conda "GitHub"\)](#)

* [\[!\[Element\]\(../../../../../_static/element_logo.svg\)\]\(http://bit.ly/conda-chat-room "Element"\)](#)

* [\[__ Discourse\]\(https://conda.discourse.group/ "Discourse"\)](#)

Navigation

- * [User guide](../././././user-guide/index.html) __
- * [Getting started with conda](../././././user-guide/getting-started.html)
- * [Installing conda](../././././user-guide/install/index.html) __
 - * [Installing on Windows](../././././user-guide/install/windows.html)
 - * [Installing on macOS](../././././user-guide/install/macos.html)
 - * [Installing on Linux](../././././user-guide/install/linux.html)
 - * [RPM and Debian Repositories for Miniconda](../././././user-guide/install/rpm-debian.html)
- * [Tasks](../././././user-guide/tasks/index.html) __
 - * [Managing conda](../././././user-guide/tasks/manage-conda.html)
 - * [Managing environments](../././././user-guide/tasks/manage-environments.html)
 - * [Managing channels](../././././user-guide/tasks/manage-channels.html)
 - * [Managing packages](../././././user-guide/tasks/manage-pkgs.html)
 - * [Managing Python](../././././user-guide/tasks/manage-python.html)
 - * [Managing virtual packages](../././././user-guide/tasks/manage-virtual.html)
 - * [Creating custom channels](../././././user-guide/tasks/create-custom-channels.html)
 - * [Creating projects](../././././user-guide/tasks/creating-projects.html)
 - * [Viewing command-line help](../././././user-guide/tasks/view-command-line-help.html)
- * [Configuration](../././././user-guide/configuration/index.html) __
 - * [Using the .condarc conda configuration file](../././././user-guide/configuration/use-condarc.html)
 - * [Settings](../././././user-guide/configuration/settings.html)
 - * [Administering a multi-user conda installation](../././././user-guide/configuration/admin-multi-user-install.html)
 - * [Mirroring channels](../././././user-guide/configuration/mirroring.html)
 - * [Disabling SSL verification](../././././user-guide/configuration/disable-ssl-verification.html)

- * [Using non-standard certificates](../../../../../user-guide/configuration/non-standard-certs.html)
- * [Using Custom Locations for Environment and Package Cache](../../../../../user-guide/configuration/custom-env-and-pkg-locations.html)
- * [Improving interoperability with pip](../../../../../user-guide/configuration/pip-interoperability.html)
- * [Using the free channel](../../../../../user-guide/configuration/free-channel.html)
- * [Concepts](../../../../../user-guide/concepts/index.html) __
- * [Commands](../../../../../user-guide/concepts/conda-commands.html)
- * [Packages](../../../../../user-guide/concepts/packages.html)
- * [Package specification](../../../../../user-guide/concepts/pkg-specs.html)
- * [Package search and install specifications](../../../../../user-guide/concepts/pkg-search.html)
- * [Channels](../../../../../user-guide/concepts/channels.html)
- * [Environments](../../../../../user-guide/concepts/environments.html)
- * [Installing with conda](../../../../../user-guide/concepts/installing-with-conda.html)
- * [Performance](../../../../../user-guide/concepts/conda-performance.html)
- * [Conda for data scientists](../../../../../user-guide/concepts/data-science.html)
- * [Plugins](../../../../../user-guide/concepts/conda-plugins.html)
- * [Troubleshooting](../../../../../user-guide/troubleshooting.html)
- * [Cheatsheet](../../../../../user-guide/cheatsheet.html)
- * [Configuration](../../../../../configuration.html)
- * [Commands](../../../../../commands/index.html) __
- * [``conda clean``](../../../../../commands/clean.html)
- * [``conda compare``](../../../../../commands/compare.html)
- * [``conda config``](../../../../../commands/config.html)
- * [``conda create``](../../../../../commands/create.html)
- * [``conda doctor``](../../../../../commands/doctor.html)
- * [``conda env``](../../../../../commands/env/index.html) __
- * [``conda env config``](../../../../../commands/env/config/index.html) __

- * [`conda env config vars`](../.../commands/env/config/vars/index.html) __
- * [`conda env config vars list`](../.../commands/env/config/vars/list.html)
- * [`conda env config vars set`](../.../commands/env/config/vars/set.html)
- * [`conda env config vars unset`](../.../commands/env/config/vars/unset.html)
- * [`conda env create`](../.../commands/env/create.html)
- * [`conda env export`](../.../commands/env/export.html)
- * [`conda env list`](../.../commands/env/list.html)
- * [`conda env remove`](../.../commands/env/remove.html)
- * [`conda env update`](../.../commands/env/update.html)
- * [`conda info`](../.../commands/info.html)
- * [`conda init`](../.../commands/init.html)
- * [`conda install`](../.../commands/install.html)
- * [`conda list`](../.../commands/list.html)
- * [`conda notices`](../.../commands/notices.html)
- * [`conda package`](../.../commands/package.html)
- * [`conda remove`](../.../commands/remove.html)
- * [`conda rename`](../.../commands/rename.html)
- * [`conda run`](../.../commands/run.html)
- * [`conda search`](../.../commands/search.html)
- * [`conda update`](../.../commands/update.html)
- * [Release notes](../.../release-notes.html)
- * [Glossary](../.../glossary.html)
- * [Developer guide](../.../index.html) __
- * [Architecture](../.../architecture.html)
- * [Contributing to conda](../.../contributing.html)
- * [Development Environment](../.../development-environment.html)
- * [Deep dives](../.../deep-dives/index.html) __

- * [`conda install``](../../deep-dives/install.html)
- * [`conda init`` and `conda activate``](../../deep-dives/activation.html)
- * [`conda config`` and context](../../deep-dives/context.html)
- * [Solvers](../../deep-dives/solvers.html)
- * [Logging](../../deep-dives/logging.html)
- * [Writing Tests](../../writing-tests/index.html) __
- * [Integration Tests](../../writing-tests/integration-tests.html)
- * [Deprecations](../../deprecations.html)
- * [Releasing](../../releasing.html)
- * [Plugins](../../plugins/index.html) __
 - * [Auth Handlers](../../plugins/auth_handlers.html)
 - * [Health Checks](../../plugins/health_checks.html)
 - * [Request Headers](../../plugins/request_headers.html)
 - * [Post-commands](../../plugins/post_commands.html)
 - * [Pre-commands](../../plugins/pre_commands.html)
 - * [Reporter Backends](../../plugins/reporter_backends.html)
 - * [Settings](../../plugins/settings.html)
 - * [Solvers](../../plugins/solvers.html)
 - * [Subcommands](../../plugins/subcommands.html)
 - * [Virtual Packages](../../plugins/virtual_packages.html)
- * [Specifications](../../specs/index.html) __
 - * [Technical specification: solver state](../../specs/solver-state.html)
- * [API](../../api.html) __
 - * [`conda``](../index.html) __
 - * [`__main__``](../__main__/index.html)
 - * [`_vendor``](../_vendor/index.html) __
 - * [`frozendict``](../_vendor/frozendict/index.html)

- * [`_version``](../_version/index.html)
- * [`activate``](../activate/index.html)
- * [`api``](../api/index.html)
- * [`auxlib``](../auxlib/index.html) __
 - * [`collection``](../auxlib/collection/index.html)
 - * [`compat``](../auxlib/compat/index.html)
 - * [`decorators``](../auxlib/decorators/index.html)
 - * [`entity``](../auxlib/entity/index.html)
 - * [`exceptions``](../auxlib/exceptions/index.html)
 - * [`ish``](../auxlib/ish/index.html)
 - * [`logz``](../auxlib/logz/index.html)
 - * [`type_coercion``](../auxlib/type_coercion/index.html)
- * [`base``](../base/index.html) __
 - * [`constants``](../base/constants/index.html)
 - * [`context``](../base/context/index.html)
- * [`cli``](../cli/index.html) __
 - * [`actions``](../cli/actions/index.html)
 - * [`common``](../cli/common/index.html)
 - * [`conda_argparse``](../cli/conda_argparse/index.html)
 - * [`find_commands``](../cli/find_commands/index.html)
 - * [`helpers``](../cli/helpers/index.html)
 - * [`install``](../cli/install/index.html)
 - * [`main``](../cli/main/index.html)
 - * [`main_clean``](../cli/main_clean/index.html)
 - * [`main_commands``](../cli/main_commands/index.html)
 - * [`main_compare``](../cli/main_compare/index.html)
 - * [`main_config``](../cli/main_config/index.html)

- * [main_create](../../cli/main_create/index.html)
- * [main_env](../../cli/main_env/index.html)
- * [main_env_config](../../cli/main_env_config/index.html)
- * [main_env_create](../../cli/main_env_create/index.html)
- * [main_env_export](../../cli/main_env_export/index.html)
- * [main_env_list](../../cli/main_env_list/index.html)
- * [main_env_remove](../../cli/main_env_remove/index.html)
- * [main_env_update](../../cli/main_env_update/index.html)
- * [main_env_vars](../../cli/main_env_vars/index.html)
- * [main_export](../../cli/main_export/index.html)
- * [main_info](../../cli/main_info/index.html)
- * [main_init](../../cli/main_init/index.html)
- * [main_install](../../cli/main_install/index.html)
- * [main_list](../../cli/main_list/index.html)
- * [main_mock_activate](../../cli/main_mock_activate/index.html)
- * [main_mock_deactivate](../../cli/main_mock_deactivate/index.html)
- * [main_notices](../../cli/main_notices/index.html)
- * [main_package](../../cli/main_package/index.html)
- * [main_pip](../../cli/main_pip/index.html)
- * [main_remove](../../cli/main_remove/index.html)
- * [main_rename](../../cli/main_rename/index.html)
- * [main_run](../../cli/main_run/index.html)
- * [main_search](../../cli/main_search/index.html)
- * [main_update](../../cli/main_update/index.html)
- * [python_api](../../cli/python_api/index.html)
- * [common](../index.html) __
- * [_logic](../_logic/index.html)

- * [`_os``](../_os/index.html) __
- * [`linux``](../_os/linux/index.html)
- * [`osx``](../_os/osx/index.html)
- * [`unix``](../_os/unix/index.html)
- * [`windows``](../_os/windows/index.html)
- * [`compat``](../compat/index.html)
- * [`configuration``](../configuration/index.html)
- * [`constants``](../constants/index.html)
- * [`disk``](../disk/index.html)
- * [`io``](../io/index.html)
- * [`iterators``](../iterators/index.html)
- * `logic``
- * [`path``](../path/index.html) __
- * [`_cygpath``](../path/_cygpath/index.html)
- * [`directories``](../path/directories/index.html)
- * [`python``](../path/python/index.html)
- * [`windows``](../path/windows/index.html)
- * [`pkg_formats``](../pkg_formats/index.html) __
- * [`python``](../pkg_formats/python/index.html)
- * [`serialize``](../serialize/index.html)
- * [`signals``](../signals/index.html)
- * [`toposort``](../toposort/index.html)
- * [`url``](../url/index.html)
- * [`core``](.././core/index.html) __
- * [`envs_manager``](.././core/envs_manager/index.html)
- * [`index``](.././core/index/index.html)
- * [`initialize``](.././core/initialize/index.html)

- * [link`](../../core/link/index.html)
- * [package_cache_data`](../../core/package_cache_data/index.html)
- * [path_actions`](../../core/path_actions/index.html)
- * [portability`](../../core/portability/index.html)
- * [prefix_data`](../../core/prefix_data/index.html)
- * [solve`](../../core/solve/index.html)
- * [subdir_data`](../../core/subdir_data/index.html)
- * [deprecations`](../../deprecations/index.html)
- * [env`](../../env/index.html) ____
- * [env`](../../env/env/index.html)
- * [installers`](../../env/installers/index.html) ____
- * [base`](../../env/installers/base/index.html)
- * [conda`](../../env/installers/conda/index.html)
- * [pip`](../../env/installers/pip/index.html)
- * [pip_util`](../../env/pip_util/index.html)
- * [specs`](../../env/specs/index.html) ____
- * [binstar`](../../env/specs/binstar/index.html)
- * [requirements`](../../env/specs/requirements/index.html)
- * [yaml_file`](../../env/specs/yaml_file/index.html)
- * [exception_handler`](../../exception_handler/index.html)
- * [exceptions`](../../exceptions/index.html)
- * [exports`](../../exports/index.html)
- * [gateways`](../../gateways/index.html) ____
- * [anaconda_client`](../../gateways/anaconda_client/index.html)
- * [connection`](../../gateways/connection/index.html) ____
- * [adapters`](../../gateways/connection/adapters/index.html) ____
- * [ftp`](../../gateways/connection/adapters/ftp/index.html)

- * [`http`](../../gateways/connection/adapters/http/index.html)
- * [`localfs`](../../gateways/connection/adapters/localfs/index.html)
- * [`s3`](../../gateways/connection/adapters/s3/index.html)
- * [`download`](../../gateways/connection/download/index.html)
- * [`session`](../../gateways/connection/session/index.html)
- * [`disk`](../../gateways/disk/index.html) __
- * [`create`](../../gateways/disk/create/index.html)
- * [`delete`](../../gateways/disk/delete/index.html)
- * [`link`](../../gateways/disk/link/index.html)
- * [`lock`](../../gateways/disk/lock/index.html)
- * [`permissions`](../../gateways/disk/permissions/index.html)
- * [`read`](../../gateways/disk/read/index.html)
- * [`test`](../../gateways/disk/test/index.html)
- * [`update`](../../gateways/disk/update/index.html)
- * [`logging`](../../gateways/logging/index.html)
- * [`repodata`](../../gateways/repodata/index.html) __
- * [`jlap`](../../gateways/repodata/jlap/index.html) __
- * [`core`](../../gateways/repodata/jlap/core/index.html)
- * [`fetch`](../../gateways/repodata/jlap/fetch/index.html)
- * [`interface`](../../gateways/repodata/jlap/interface/index.html)
- * [`lock`](../../gateways/repodata/lock/index.html)
- * [`subprocess`](../../gateways/subprocess/index.html)
- * [`history`](../../history/index.html)
- * [`instructions`](../../instructions/index.html)
- * [`misc`](../../misc/index.html)
- * [`models`](../../models/index.html) __
- * [`channel`](../../models/channel/index.html)

- * [`dist``](../../models/dist/index.html)
- * [`enums``](../../models/enums/index.html)
- * [`leased_path_entry``](../../models/leased_path_entry/index.html)
- * [`match_spec``](../../models/match_spec/index.html)
- * [`package_info``](../../models/package_info/index.html)
- * [`prefix_graph``](../../models/prefix_graph/index.html)
- * [`records``](../../models/records/index.html)
- * [`version``](../../models/version/index.html)
- * [`notices``](../../notices/index.html) __
- * [`cache``](../../notices/cache/index.html)
- * [`core``](../../notices/core/index.html)
- * [`fetch``](../../notices/fetch/index.html)
- * [`types``](../../notices/types/index.html)
- * [`views``](../../notices/views/index.html)
- * [`plan``](../../plan/index.html)
- * [`plugins``](../../plugins/index.html) __
- * [`hookspec``](../../plugins/hookspec/index.html)
- * [`manager``](../../plugins/manager/index.html)
- * [`post_solves``](../../plugins/post_solves/index.html) __
 - * [`signature_verification``](../../plugins/post_solves/signature_verification/index.html)
- * [`reporter_backends``](../../plugins/reporter_backends/index.html) __
 - * [`console``](../../plugins/reporter_backends/console/index.html)
 - * [`json``](../../plugins/reporter_backends/json/index.html)
- * [`solvers``](../../plugins/solvers/index.html)
- * [`subcommands``](../../plugins/subcommands/index.html) __
 - * [`doctor``](../../plugins/subcommands/doctor/index.html) __
 - * [`health_checks``](../../plugins/subcommands/doctor/health_checks/index.html)

- * [`types``](../../plugins/types/index.html)
- * [`virtual_packages``](../../plugins/virtual_packages/index.html) __
 - * [`archspeg``](../../plugins/virtual_packages/archspeg/index.html)
 - * [`conda``](../../plugins/virtual_packages/conda/index.html)
 - * [`cuda``](../../plugins/virtual_packages/cuda/index.html)
 - * [`freebsd``](../../plugins/virtual_packages/freebsd/index.html)
 - * [`linux``](../../plugins/virtual_packages/linux/index.html)
 - * [`osx``](../../plugins/virtual_packages/osx/index.html)
 - * [`windows``](../../plugins/virtual_packages/windows/index.html)
- * [`reporters``](../../reporters/index.html)
- * [`resolve``](../../resolve/index.html)
- * [`testing``](../../testing/index.html) __
 - * [`cases``](../../testing/cases/index.html)
 - * [`fixtures``](../../testing/fixtures/index.html)
 - * [`gateways``](../../testing/gateways/index.html) __
 - * [`fixtures``](../../testing/gateways/fixtures/index.html)
 - * [`helpers``](../../testing/helpers/index.html)
 - * [`integration``](../../testing/integration/index.html)
 - * [`notices``](../../testing/notices/index.html) __
 - * [`fixtures``](../../testing/notices/fixtures/index.html)
 - * [`helpers``](../../testing/notices/helpers/index.html)
 - * [`solver_helpers``](../../testing/solver_helpers/index.html)
- * [`trust``](../../trust/index.html) __
 - * [`constants``](../../trust/constants/index.html)
 - * [`signature_verification``](../../trust/signature_verification/index.html)
- * [`utils``](../../utils/index.html)
- * [`conda_env``](../../conda_env/index.html) __

```
* [ `cli`](../../conda_env/cli/index.html)
```

```
* [ `installers`](../../conda_env/installers/index.html)
```

```
* [ ____](../../index.html)
```

```
* [Developer guide](../../index.html)
```

```
* ____
```

```
* [ `common`](../index.html)
```

```
*
```

```
# `logic`#
```

The basic idea to nest logical expressions is instead of trying to denest things via distribution, we add new variables. So if we have some logical expression `expr`, we replace it with `x` and add `expr <-> x` to the clauses, where `x` is a new variable, and `expr <-> x` is recursively evaluated in the same way, so that the final clauses are ORs of atoms.

To use this, create a new `Clauses` object with the max var, for instance, if you already have `[[1, 2, -3]]`, you would use `C = Clause(3)`. All functions return a new literal, which represents that function, or `True` or `False` if the expression can be resolved fully. They may also add new clauses to `C.clauses`, which will then be delivered to the SAT solver.

All functions take atoms as arguments (an atom is an integer, representing a literal or a negated literal, or boolean constants `True` or `False`; that is, it is the callers' responsibility to do the conversion of expressions recursively. This is done because we do not have data structures representing

the various logical classes, only atoms.

The polarity argument can be set to True or False if you know that the literal being used will only be used in the positive or the negative, respectively (e.g., you will only use x , not $-x$). This will generate fewer clauses. It is probably best if you do not take advantage of this directly, but rather through the Require and Prevent functions.

Classes#

`Clauses` |

---|---

Functions#

`minimal_unsatisfiable_subset` (clauses, sat, explicit_specs) | Given a set of clauses, find a minimal unsatisfiable subset (an

---|---

Attributes#

`TRUE` |

---|---

`FALSE` |

`PycoSatSolver` |

`PyCryptoSatSolver` |

`PySatSolver` |

TRUE#

FALSE#

PycoSatSolver _ = 'pocosat' _#

PyCryptoSatSolver _ = 'pccryptosat' _#

PySatSolver _ = 'pysat' _#

_class _Clauses(_m =0_, _sat_solver =PycoSatSolver_)#

_property _m#

`_property _unsat#`

`get_clause_count()#`

`as_list()#`

`_check_variable(_variable_)#`

`_check_literal(_literal_)#`

`add_clause(_clause_)#`

`add_clauses(_clauses_)#`

name_var(_m_ , _name_)#

new_var(_name =None_)#

from_name(_name_)#

from_index(_m_)#

_assign(_vals_ , _name =None_)#

_convert(_x_)#

`_eval(_func_ , _args_ , _no_literal_args_ , _polarity_ , _name_)#`

`Prevent(_what_ , _* args_)#`

`Require(_what_ , _* args_)#`

`Not(_x_ , _polarity =None_ , _name =None_)#`

`And(_f_ , _g_ , _polarity =None_ , _name =None_)#`

`Or(_f_ , _g_ , _polarity =None_ , _name =None_)#`

`Xor(_f_ , _g_ , _polarity =None_ , _name =None_)#`

```
ITE(_c_ , _t_ , _f_ , _polarity =None_, _name =None_)#
```

If c Then t Else f.

In this function, if any of c, t, or f are True and False the resulting expression is resolved.

```
All(_iter_ , _polarity =None_, _name =None_)#
```

```
Any(_vals_ , _polarity =None_, _name =None_)#
```

```
AtMostOne_NSQ(_vals_ , _polarity =None_, _name =None_)#
```

```
AtMostOne_BDD(_vals_ , _polarity =None_, _name =None_)#
```

```
AtMostOne(_vals_ , _polarity =None_, _name =None_)#
```

ExactlyOne_NSQ(_vals_ , _polarity =None_, _name =None_)#

ExactlyOne_BDD(_vals_ , _polarity =None_, _name =None_)#

ExactlyOne(_vals_ , _polarity =None_, _name =None_)#

LinearBound(_equation_ , _lo_ , _hi_ , _preprocess =True_, _polarity =None_,
name =None)#

sat(_additional =None_, _includelf =False_, _names =False_, _limit =0_)#

Calculate a SAT solution for the current clause set.

Returned is the list of those solutions. When the clauses are unsatisfiable,

an empty list is returned.

```
itersolve(_constraints =None_, _m =None_)#
```

```
minimize(_objective_ , _bestsol =None_, _trymax =False_)#
```

```
minimal_unsatisfiable_subset(_clauses_ , _sat_ , _explicit_specs_)#
```

Given a set of clauses, find a minimal unsatisfiable subset (an unsatisfiable core)

A set is a minimal unsatisfiable subset if no proper subset is unsatisfiable.

A set of clauses may have many minimal unsatisfiable subsets of different sizes.

sat should be a function that takes a tuple of clauses and returns True if the clauses are satisfiable and False if they are not. The algorithm will work with any order-reversing function (reversing the order of subset and the order $\text{False} < \text{True}$), that is, any function where $(A \leq B)$ iff $(\text{sat}(B) \leq \text{sat}(A))$, where $A \leq B$ means A is a subset of B and $\text{False} < \text{True}$).

___On this page

- * Classes

- * Functions

- * Attributes

- * `TRUE`

- * `FALSE`

- * `PycoSatSolver`

- * `PyCryptoSatSolver`

- * `PySatSolver`

- * `Clauses`

- * `Clauses.m`

- * `Clauses.unsat`

- * `Clauses.get_clause_count()`

- * `Clauses.as_list()`

- * `Clauses._check_variable()`

- * `Clauses._check_literal()`

- * `Clauses.add_clause()`

- * `Clauses.add_clauses()`

- * `Clauses.name_var()`

- * `Clauses.new_var()`

- * `Clauses.from_name()`

- * `Clauses.from_index()`

- * `Clauses._assign()`

- * `Clauses._convert()`

- * `Clauses._eval()`

- * `Clauses.Prevent()`

- * `Clauses.Require()`
- * `Clauses.Not()`
- * `Clauses.And()`
- * `Clauses.Or()`
- * `Clauses.Xor()`
- * `Clauses.ITE()`
- * `Clauses.All()`
- * `Clauses.Any()`
- * `Clauses.AtMostOne_NSQ()`
- * `Clauses.AtMostOne_BDD()`
- * `Clauses.AtMostOne()`
- * `Clauses.ExactlyOne_NSQ()`
- * `Clauses.ExactlyOne_BDD()`
- * `Clauses.ExactlyOne()`
- * `Clauses.LinearBound()`
- * `Clauses.sat()`
- * `Clauses.itsolve()`
- * `Clauses.minimize()`
- * `minimal_unsatisfiable_subset()`

[[__Edit on GitHub](https://github.com/conda/conda/edit/main/docs/source/dev-guide/api/conda/common/logic/index.rst)](https://github.com/conda/conda/edit/main/docs/source/dev-guide/api/conda/common/logic/index.rst)

[[__Show Source](https://github.com/conda/conda/blob/main/docs/source/dev-guide/api/conda/common/logic/index.rst.txt)](../../../../../_sources/dev-guide/api/conda/common/logic/index.rst.txt)

Created using [Sphinx](<https://www.sphinx-doc.org/>) 7.4.7.

[Analytics Dashboard __](<https://docs-conda-io.goatcounter.com> "Analytics Dashboard")

Built with the [PyData Sphinx Theme](<https://pydata-sphinx-theme.readthedocs.io/en/stable/index.html>) 0.15.4.