diffblue
AI for Code

CBMC Culture & Process

Thomas Given-Wilson
2023/07/26

Structure

- CBMC Overview
- Pragmatic Development
- Review & Discuss 4 PRs
- Code structure w.r.t. CODEOWNERS
- Don’t Die
- Communication
- Releases

CBMC in a Nutshell

- Initially academic tool for checking ANSI-C Programs
- Open source
- Applied for industrial uses
- Extended to other languages
- Extended for usability
- Extended to various cprover tools
- Now >21 years old (able to drink even in the US)

CBMC Tensions

- Academia versus Industry
- Experimentation versus Stability
- CBMC core versus Peripherals
- Users versus Developers

Pragmatic Development

1. Is this experimental or stable CBMC?
   a. Experimental features can be less well tested and reviewed
   b. Stable CBMC needs a lot of tests and scrutiny
2. Is this CBMC core or peripheral?
   a. CBMC core binary and structures are very hard to alter
   b. Peripheral/new tools are quite flexible
3. Who is this development for?
   a. Users of Cprover tools?
   b. Developers of/with Cprover tools?

Enough Talk, Let’s Look 4 at PRs

- All PRs we’ll look at have been merged (so they’re all “successful”)
- All have good and bad points
- Most are from people here (or at least from AWS and Diffblue)

PR - SYNTHESIZER: Add enumerative loop invariant synthesizer

https://github.com/diffblue/cbmc/pull/7430

PR - SYNTHESIZER: Add enumerative loop invariant synthesizer … Thoughts

Good

- Title
- PR Body Text
- Regression tests
- In code comments

Could improve

- One giant commit
- Many code regions/CODEOWNERS

PR - Mark CBMC version 5.80.0

https://github.com/diffblue/cbmc/pull/7637

PR - Mark CBMC version 5.80.0 … Thoughts

Good

- PR Body Text

Could improve

- Title and body don’t match
- Regression tests depend on minor version?!

PR - Add new function to inline a list of calls in a goto_program

https://github.com/diffblue/cbmc/pull/7550

PR - Add new function to inline a list of calls in a goto_program … Thoughts

Good

- Title
- PR Body Text
- Split new functionality required into small/local change

Could improve

- One commit (but not too big)
- Missing unit tests and doxygen (on first raising PR)

PR - Replace requires with c_requires

https://github.com/diffblue/cbmc/pull/7698

PR - Replace requires with c_requires … Thoughts

Good

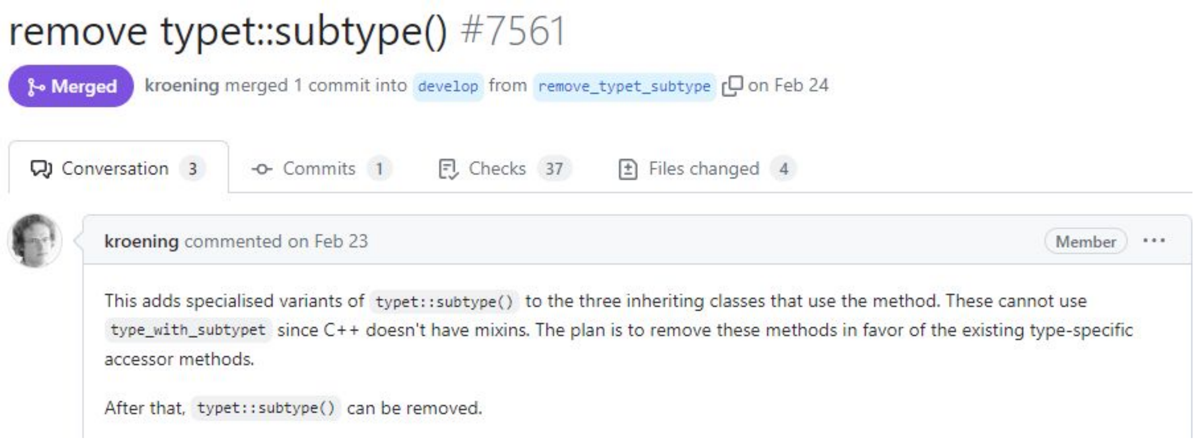
- Title and PR Body Text
- Targeted/clear scope

Could improve

- Didn’t do c_assigns, c_ensures, c_frees until prompted
- Messy commits (resolved before merging)
- Still missed some… Requires keyword is still present in code base (with regards to C++20)

Some Good PRs for Reference

- remove typet::subtype()
- Refactor and add documentation for mmio goto-program pass
- Add bvt output helper



remove typet::subtype() #7561

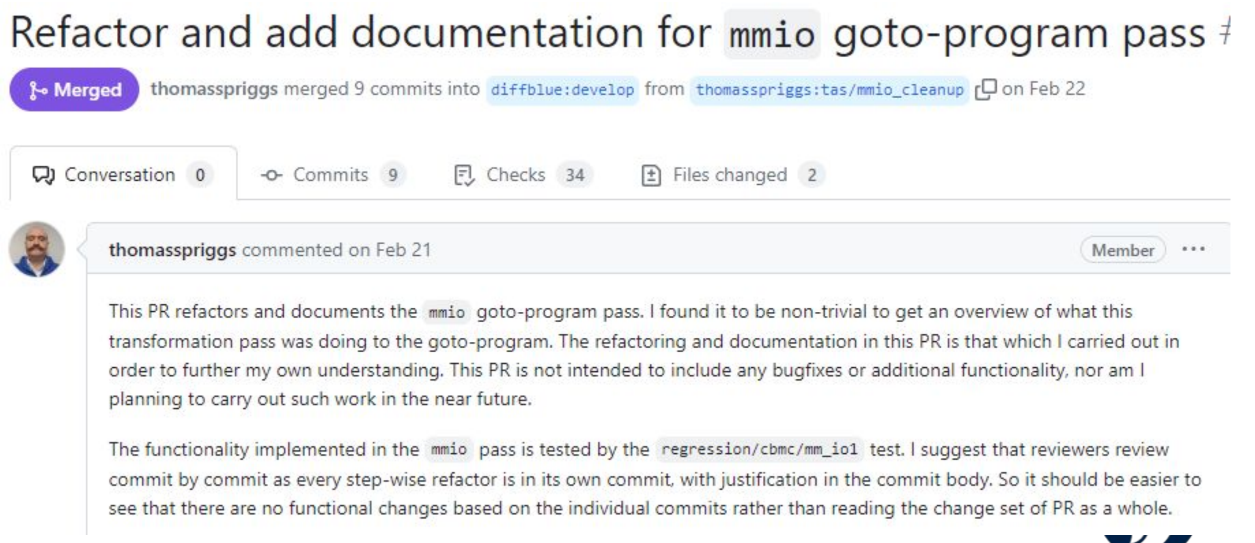
Merged kroening merged 1 commit into develop from remove_typet_subtype on Feb 24

Conversation 3 · Commits 1 · Checks 37 · Files changed 4

kroening commented on Feb 23 · Member

This adds specialised variants of `typet::subtype()` to the three inheriting classes that use the method. These cannot use `type_with_subtype` since C++ doesn't have mixins. The plan is to remove these methods in favor of the existing type-specific accessor methods.

After that, `typet::subtype()` can be removed.



Refactor and add documentation for `mmio` goto-program pass #

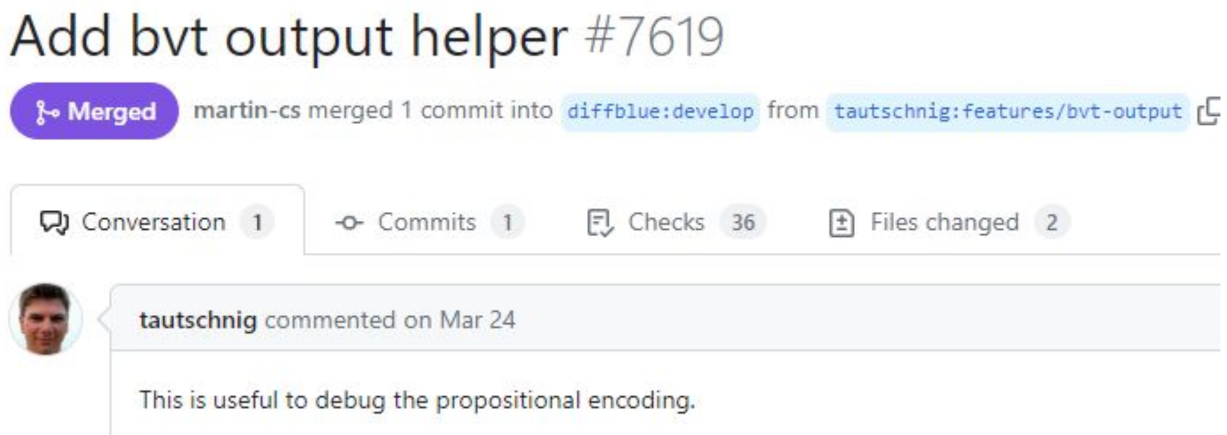
Merged thomasspriggs merged 9 commits into diffblue:develop from thomasspriggs:tas/mmio_cleanup on Feb 22

Conversation 0 · Commits 9 · Checks 34 · Files changed 2

thomasspriggs commented on Feb 21 · Member

This PR refactors and documents the `mmio` goto-program pass. I found it to be non-trivial to get an overview of what this transformation pass was doing to the goto-program. The refactoring and documentation in this PR is that which I carried out in order to further my own understanding. This PR is not intended to include any bugfixes or additional functionality, nor am I planning to carry out such work in the near future.

The functionality implemented in the `mmio` pass is tested by the `regression/cbmc/mm_io1` test. I suggest that reviewers review commit by commit as every step-wise refactor is in its own commit, with justification in the commit body. So it should be easier to see that there are no functional changes based on the individual commits rather than reading the change set of PR as a whole.



Add bvt output helper #7619

Merged martin-cs merged 1 commit into diffblue:develop from tautschnig:features/bvt-output

Conversation 1 · Commits 1 · Checks 36 · Files changed 2

tautschnig commented on Mar 24

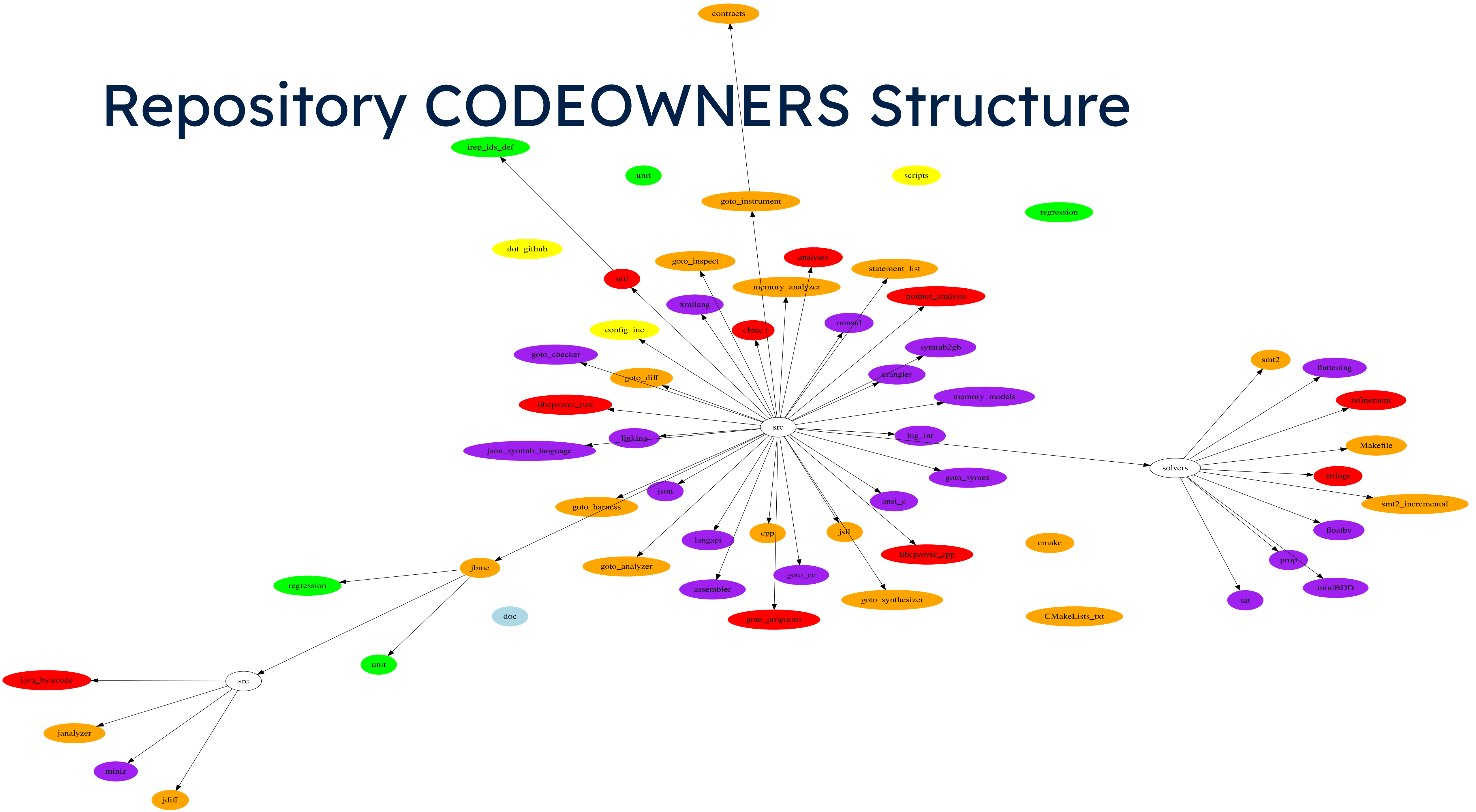
This is useful to debug the propositional encoding.

Reviewing Perspective

Mostly about how I/Diffblue look at PRs

1. Read title and body
   a. Check CI status, may ignore PR if any required CI job is failing
2. Read conversation/comments/other reviews *(maybe at the end)*
3. Go through commits
   a. Look at code/changes per commit
4. Look at overall changes
   a. Examine code, tests, etc.
5. *Maybe* build off branch and test locally/experiment.
6. Review and/or add comments *(may be CODEOWNER area limited)*

Repository CODEOWNERS Structure



Refactor or Die

Title stolen from this 5 minute talk. Refactored key points for CBMC

- Many small PRs merge exponentially faster than 1 big PR
  - Split out hooks into other code areas/CODEOWNERS
    - or at least split commits
- Understand/explain WHY you need to touch that code area
  - and make this clear in the PR
- Make your PR easy to review
- Include tests (unit and/or regression)

- … have Daniel, Michael, or Peter on board
  - or other appropriate CODEOWNERS

How to Communicate?

Challenges with CBMC contributors

- RFC?
  - can lead discussion
  - can converge on solution
  - can group together many PRs (e.g. Kani API, new SMT)
- PR discussion?
  - less likely to converge
  - easily kills a PR
  - … but can motivate competing PRs
- Documentation?
  - easier to get merged
  - likely to be lost

CBMC Releases

- Major releases only when major changes occur (very rare)
    - Implies major changes to core structure/behaviour, can break user experience
- Minor releases every two weeks
    - Time based cadence, but can drift a day or two for PRs or holidays.
- Point releases
    - Rarely done, usually for urgent PR/feature or important bug fix.
- Process (simplified):
    - Update version numbers (git branch)
    - Create release by git tag and github workflows
    - Create homebrew tap and Rust API Crate (manual/scripts)

diffblue
AI for Code

Thank you