Continuous Modernization of CPAchecker

Philipp Wendler

SoSy-Lab @ LMU Munich

CPA'25 2025-06-28

SVN+Git ⇒ Git

- Discussed last workshop, completed on 2024-10-18
- Worked well, no complaints
- Thanks everyone for help and collaboration!

Policy Review

- Work in forks discouraged
  (no problems, only 2 external MRs so far)

Policy Review

- Work in forks discouraged
  (no problems, only 2 external MRs so far)
- Work in branches suggested,
  work on `main` possible for core

Policy Review

- Work in forks discouraged
  (no problems, only 2 external MRs so far)
- Work in branches suggested,
  work on `main` possible for core
- No force pushes

Policy Review

- Work in forks discouraged
  (no problems, only 2 external MRs so far)
- Work in branches suggested,
  work on `main` possible for core
- No force pushes
- Reviewing strongly encouraged, successful?

Policy Review

- Work in forks discouraged
  (no problems, only 2 external MRs so far)
- Work in branches suggested,
  work on `main` possible for core
- No force pushes
- Reviewing strongly encouraged, successful?
- Anything else?

Mailing Lists

Restructuring in December 2024:

- Now 3 lists relevant for developers:

`ci`	BuildBot and CI
`commits`	Commit mails
`developers`	Discussion and infos for developers

- Low-traffic list `developers` strongly recommended for all occasional CPAchecker developers

Code Modernization

Concluded work:

- Get rid of switch with classic case labels (`case FOO:`), enforce use of arrow labels (`case FOO ->`)
- Rewrite many switch statements to switch expressions (but further manual improvements possible)
- Explanations in `doc/StyleGuide.md`

Code Modernization

Concluded work:

- Get rid of switch with classic case labels (`case FOO:`), enforce use of arrow labels (`case FOO ->`)
- Rewrite many switch statements to switch expressions (but further manual improvements possible)
- Explanations in `doc/StyleGuide.md`

Upcoming changes:

- Automated rewrites to pattern-matching `instanceof`
  - Preparations and prototype done
  - Only waiting for !136 to avoid conflicts

Side Note: Tooling

- Main helper: Google Error Prone
  - Suggestions for switch and pattern matching
  - Can generate patch files with suggestions for whole code base
- Honorable mention: Eclipse “Source Clean Up”

Java 21

- CPAchecker currently requires Java 17 or newer.
- Following LTS release is Java 21 from September 2023.

Benefit for example more powerful `switch`:

```
switch (exp.getExpressionType()) {
  case CSimpleType s when s.getType().isIntegerType() -> ...;
  case CSimpleType s -> ...;
  case CPointerType p -> ...;
}
```

- Allows us to get rid of many `if ... instanceof ... else if` chains.
- Allows us to simplify many uses of visitor pattern.

Switch to Java 21?

- Tool chain should support Java 21 well.
- Distributions ship Java 21, except Debian:
  not present in current stable,
  next stable with Java 21 to be released this summer
- Switches to Java 11 and 17 were even sooner
  after respective releases.
- (details in #1333)

Switch to Java 21?

- Tool chain should support Java 21 well.
- Distributions ship Java 21, except Debian:
  not present in current stable,
  next stable with Java 21 to be released this summer
- Switches to Java 11 and 17 were even sooner
  after respective releases.
- (details in #1333)

When should we require Java 21?

- right now
- make a release soon, then require Java 21
- only after Debian with Java 21 is available,
  e.g., after the usual autumn release of CPAchecker

Further Long-Term Development Plans

- Migrate nullability annotations to use JSpecify
- Fully enforce nullability annotations
- Make CPAchecker a “modularized” Java project
  that is compatible with the JPMS (#1344)
  $\Rightarrow$ Allows more uses of `sealed` and restricting reflection.

Feedback and contributions welcome!

Thank you for all your contributions and cooperation!