Coccinelle Explorer: Learning Semantic Patching Interactively

Michele MARTONE

Leibniz Supercomputing Centre, Garching near Munich, Germany

Kernel devroom at FOSDEM'25
Bruxelles, February 02, 2025

Coccinelle: a program matching and transformation engine

- own C/C++ parser, own AST
- matches language entities
- matches control-flow
- can patch what is matched


All together is better!!!
BUG
COCCINELLE
inside

make coccicheck – static code analysis

```
scripts/
└── coccinelle/
    ├── api/
    │   └── ...
    ├── free/
    │   ├── clk_put.cocci
    │   ├── devm_free.cocci
    │   ├── ifnulldev_put.cocci
    │   ├── ifnullfree.cocci
    │   ├── iounmap.cocci
    │   ├── kfree.cocci
    │   └── ...
    ├── api/
    └── ...
```

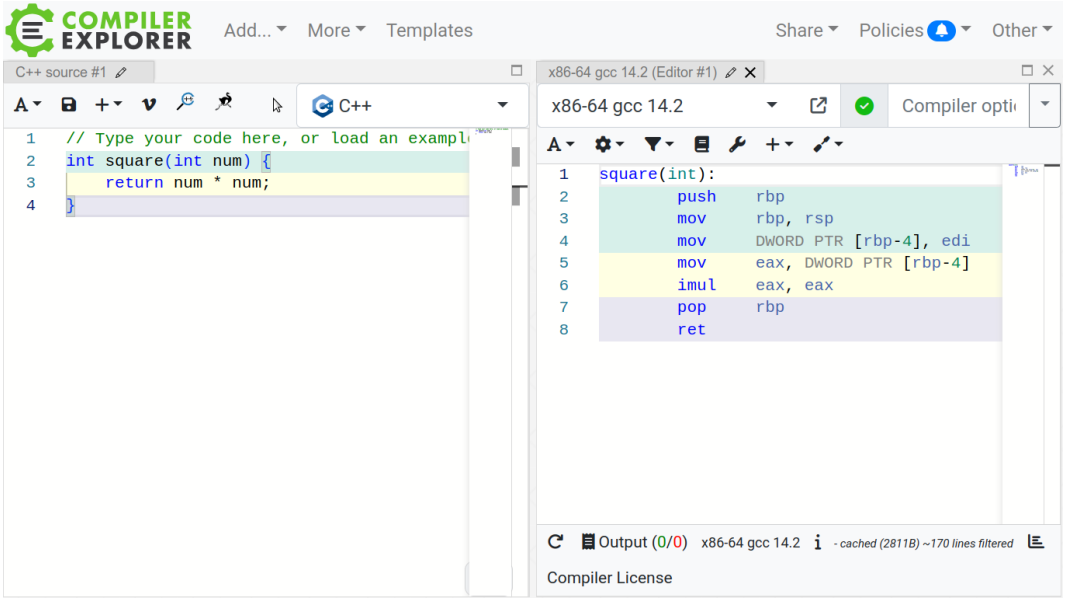
https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/

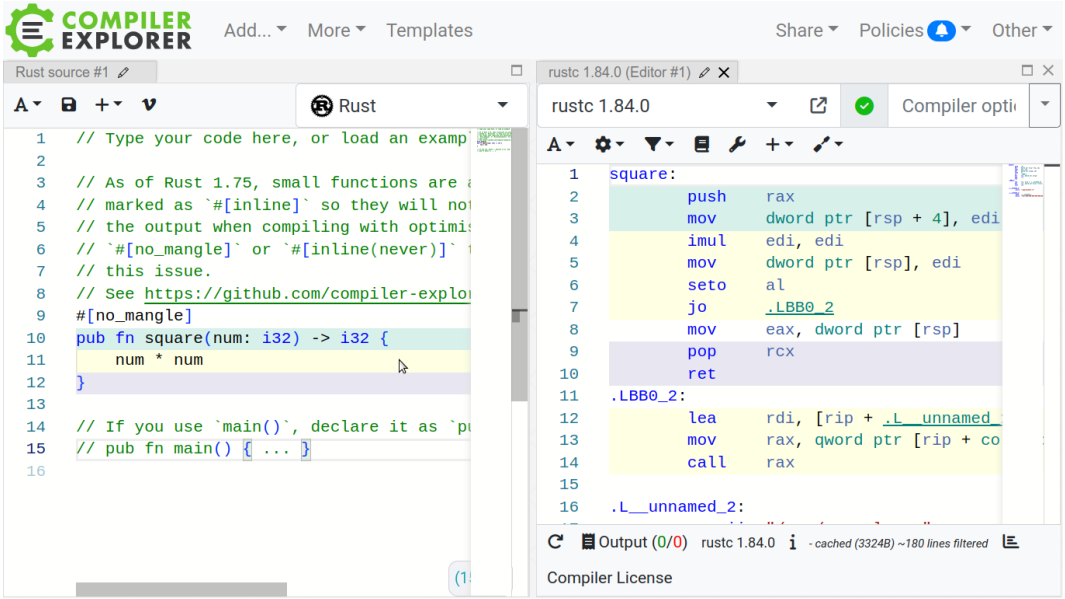
My interest

Develop HPC-oriented changes

- advanced `#pragma` manipulation (OpenMP, OpenACC, ...)
- for GPU
- C++-related
- removal of *bloat* and *cruft*

Popularization/trainings, helping HPC users


COMPILER EXPLORER
Add...
More
Templates
Share
Policies
Other
C++ source #1
C++
// Type your code here, or load an exampl
int square(int num) {
return num * num;
}
x86-64 gcc 14.2 (Editor #1)
x86-64 gcc 14.2
Compiler opti
square(int):
push rbp
mov rbp, rsp
mov DWORD PTR [rbp-4], edi
mov eax, DWORD PTR [rbp-4]
imul eax, eax
pop rbp
ret
Output (0/0)
x86-64 gcc 14.2
- cached (2811B) ~170 lines filtered
Compiler License

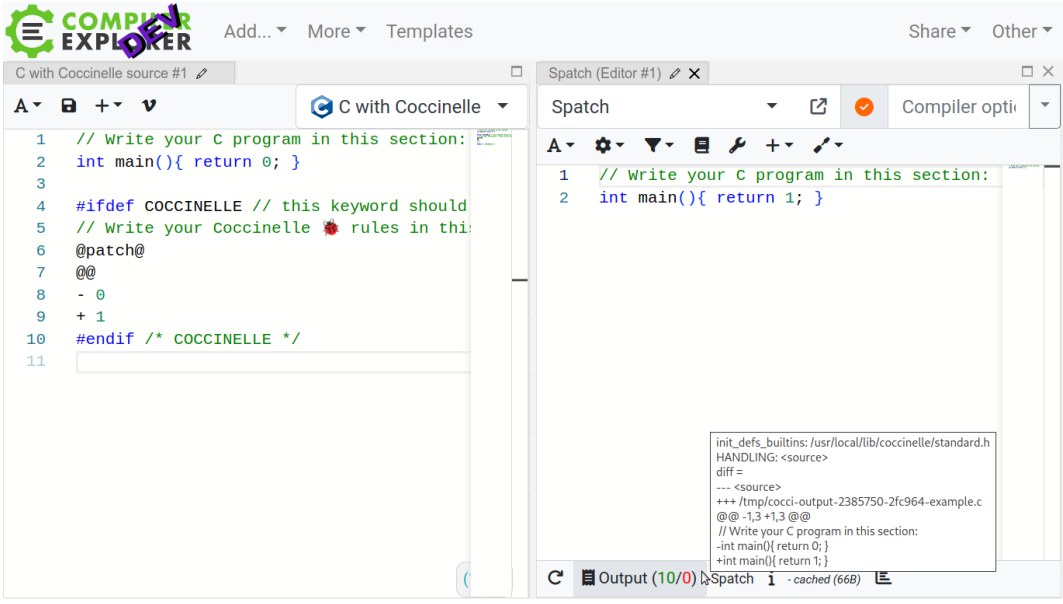

COMPILER EXPLORER
Add...
More
Templates
Share
Policies
Other
Rust source #1
Rust
rustc 1.84.0 (Editor #1)
rustc 1.84.0
Compiler opti
// Type your code here, or load an examp
// As of Rust 1.75, small functions are a
// marked as `#[inline]` so they will no
// the output when compiling with optimis
// `#[no_mangle]` or `#[inline(never)]`
// this issue.
// See https://github.com/compiler-explo
#[no_mangle]
pub fn square(num: i32) -> i32 {
num * num
}
// If you use `main()`, declare it as `pu
// pub fn main() { ... }
square:
push rax
mov dword ptr [rsp + 4], edi
imul edi, edi
mov dword ptr [rsp], edi
seto al
jo .LBB0_2
mov eax, dword ptr [rsp]
pop rcx
ret
.LBB0_2:
lea rdi, [rip + .L__unnamed_
mov rax, qword ptr [rip + co
call rax
.L__unnamed_2:
Output (0/0)
rustc 1.84.0
- cached (3324B) ~180 lines filtered
Compiler License

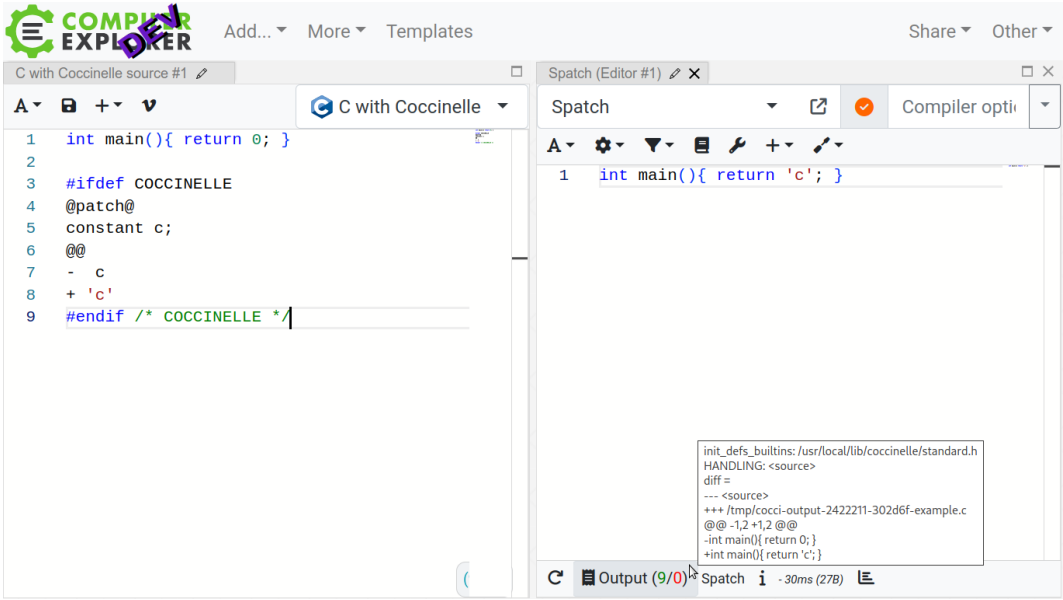
Compiler Explorer

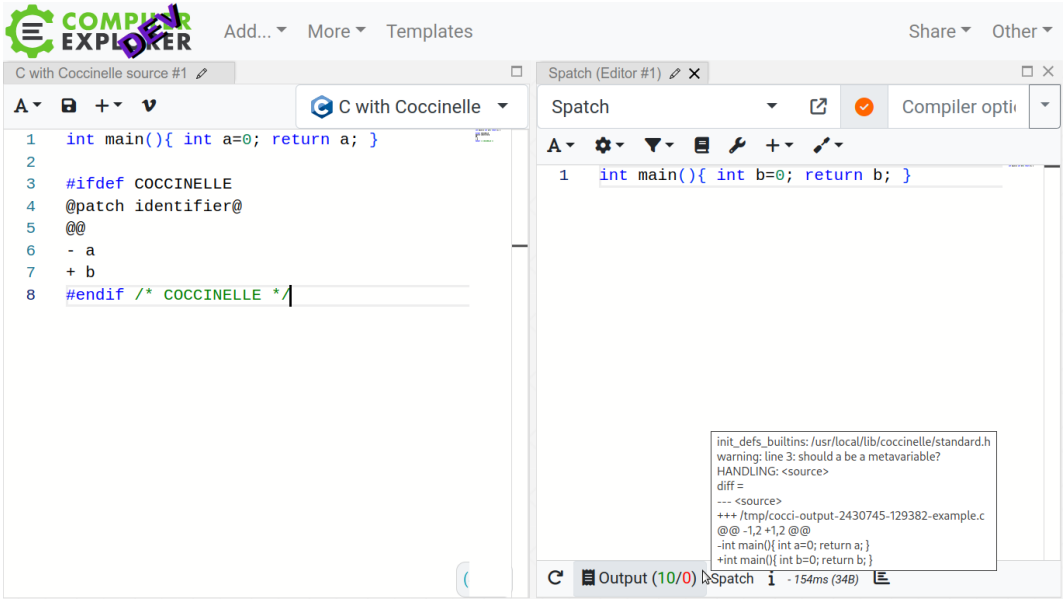
- https://godbolt.org/, online resource
- mostly TypeScript
- useful learning tool
- open to new *compilers* and *languages*

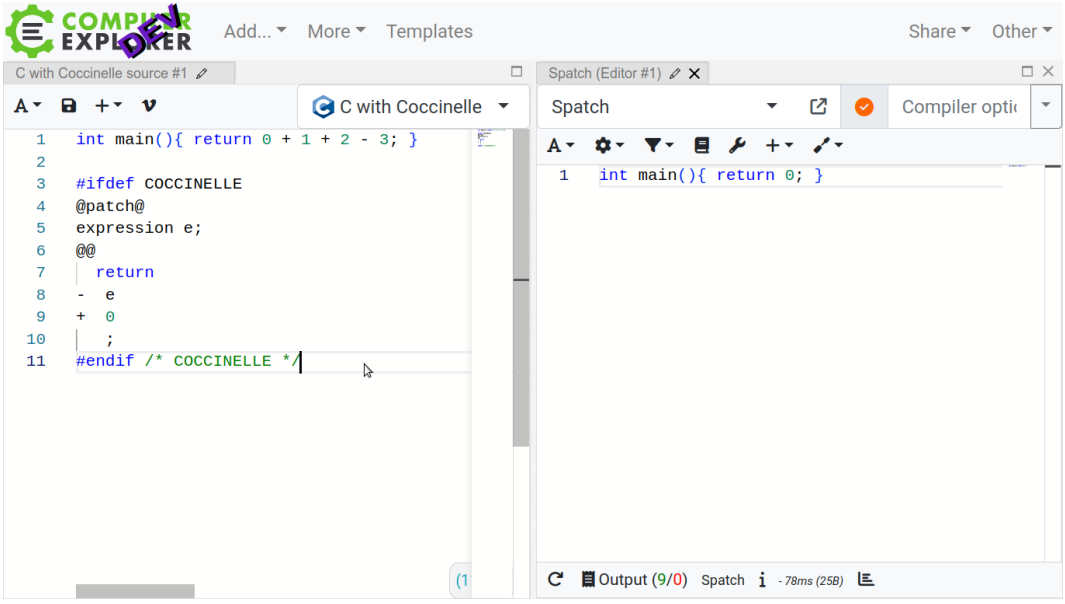
Compiler Explorer

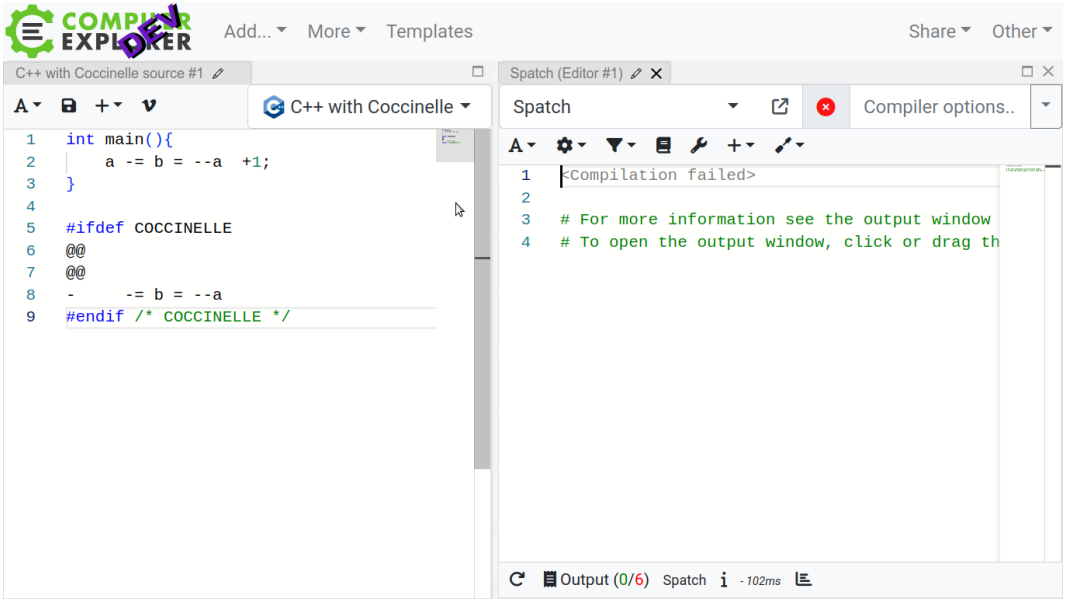
- https://godbolt.org/, online resource
- mostly TypeScript
- useful learning tool
- open to new *compilers* and *languages*
- so... why not Coccinelle?

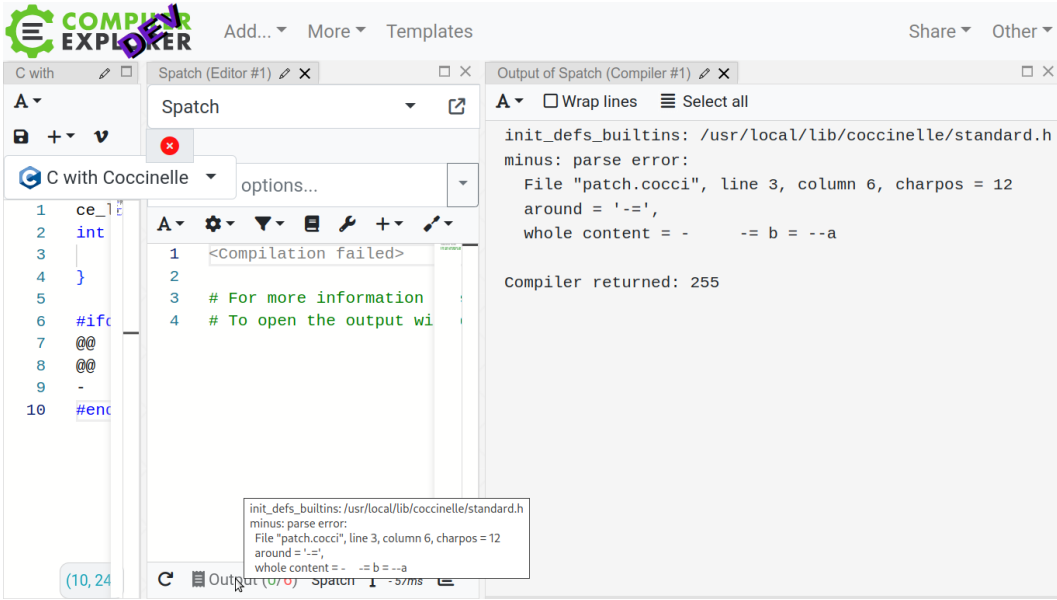

COMPILER EXPLORER
DEV
Add...
More
Templates
Share
Other
C with Coccinelle source #1
C with Coccinelle
1 // Write your C program in this section:
2 int main(){ return 0; }
3
4 #ifdef COCCINELLE // this keyword should
5 // Write your Coccinelle rules in thi
6 @patch@
7 @@
8 - 0
9 + 1
10 #endif /* COCCINELLE */
11
Spatch (Editor #1)
Spatch
Compiler opti
1 // Write your C program in this section:
2 int main(){ return 1; }
init_defs_builtins: /usr/local/lib/coccinelle/standard.h
HANDLING: <source>
diff =
--- <source>
+++ /tmp/cocci-output-2385750-2fc964-example.c
@@ -1,3 +1,3 @@
// Write your C program in this section:
-int main(){ return 0; }
+int main(){ return 1; }
Output (10/0)
Spatch
1
- cached (66B)

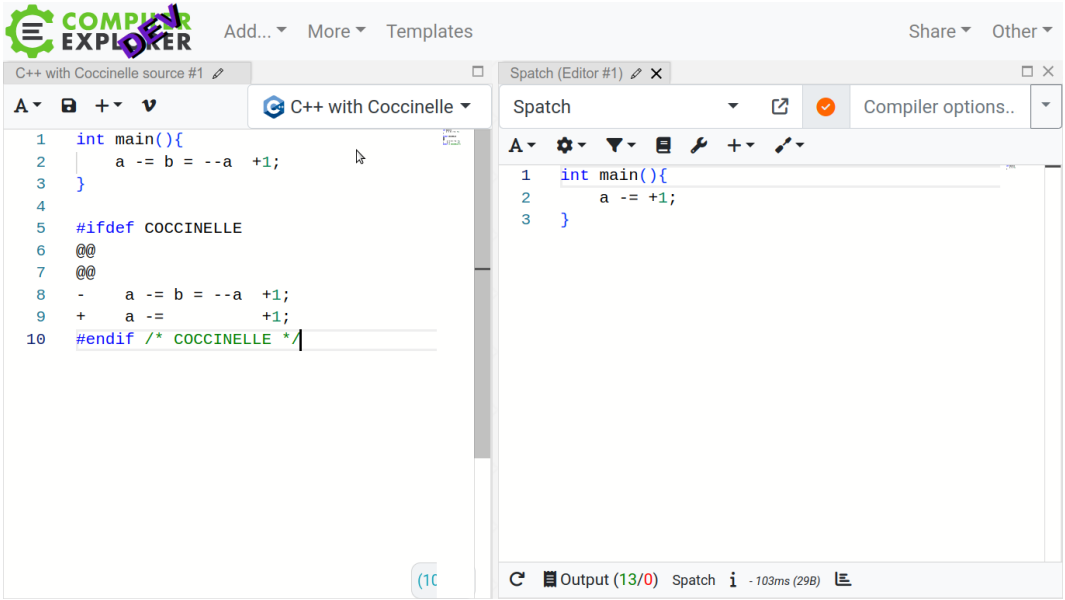

COMPILER EXPLORER DEV
Add...
More
Templates
Share
Other
C with Coccinelle source #1
C with Coccinelle
1 int main(){ return 0; }
2
3 #ifdef COCCINELLE
4 @patch@
5 constant c;
6 @@
7 - c
8 + 'c'
9 #endif /* COCCINELLE */
Spatch (Editor #1)
Spatch
Compiler opti
1 int main(){ return 'c'; }
init_defs_builtins: /usr/local/lib/coccinelle/standard.h
HANDLING: <source>
diff =
--- <source>
+++ /tmp/cocci-output-2422211-302d6f-example.c
@@ -1,2 +1,2 @@
-int main(){ return 0; }
+int main(){ return 'c'; }
Output (9/0)
Spatch
- 30ms (27B)

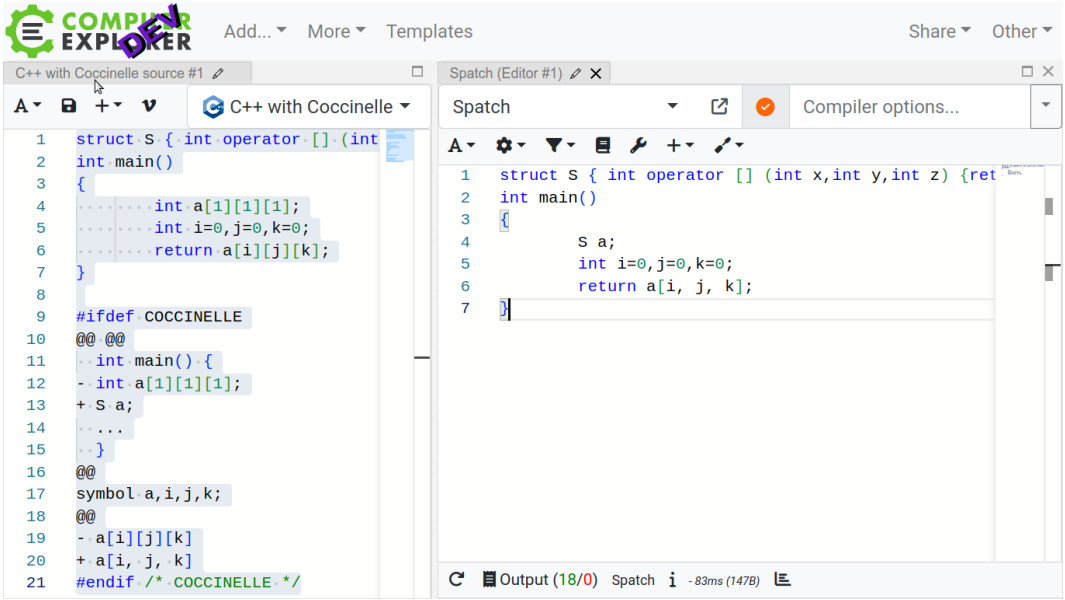

COMPILER EXPLORER
DEV
Add...
More
Templates
Share
Other
C with Coccinelle source #1
C with Coccinelle
1 int main(){ int a=0; return a; }
2
3 #ifdef COCCINELLE
4 @patch identifier@
5 @@
6 - a
7 + b
8 #endif /* COCCINELLE */
Spatch (Editor #1)
Spatch
Compiler opti
1 int main(){ int b=0; return b; }
init_defs_builtins: /usr/local/lib/coccinelle/standard.h
warning: line 3: should a be a metavariable?
HANDLING: <source>
diff =
--- <source>
+++ /tmp/cocci-output-2430745-129382-example.c
@@ -1,2 +1,2 @@
-int main(){ int a=0; return a; }
+int main(){ int b=0; return b; }
Output (10/0)
Spatch
- 154ms (34B)

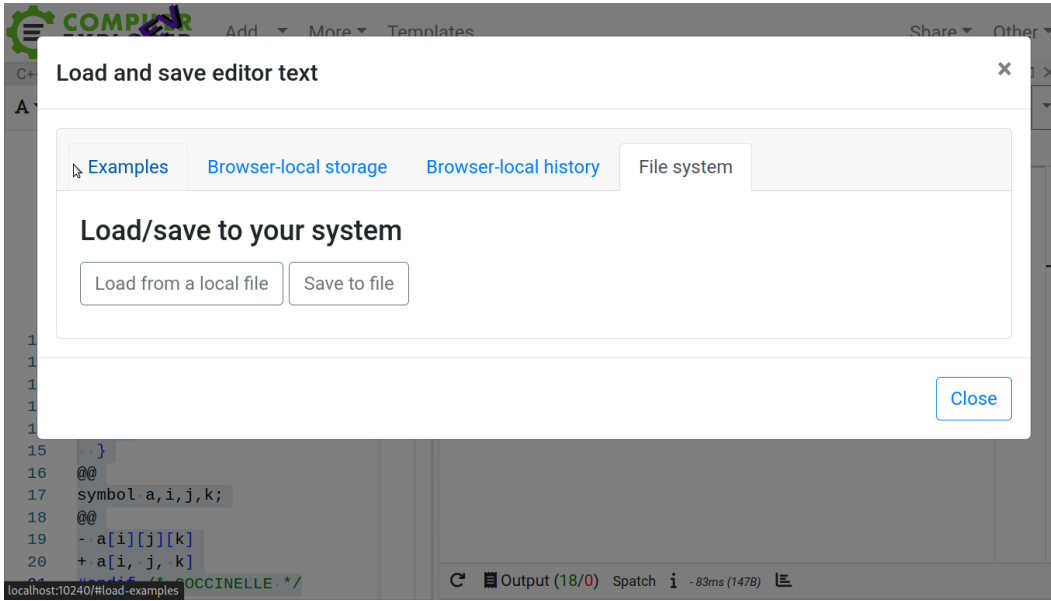

COMPILER EXPLORER
DEV
Add...
More
Templates
Share
Other
C with Coccinelle source #1
C with Coccinelle
1 int main(){ return 0 + 1 + 2 - 3; }
2
3 #ifdef COCCINELLE
4 @patch@
5 expression e;
6 @@
7 return
8 - e
9 + 0
10 ;
11 #endif /* COCCINELLE */
Spatch (Editor #1)
Spatch
Compiler opti
1 int main(){ return 0; }
Output (9/0)
Spatch
- 78ms (25B)


COMPILER EXPLORER
DEV
Add...
More
Templates
Share
Other
C++ with Coccinelle source #1
C++ with Coccinelle
1 int main(){
2 a -= b = --a +1;
3 }
4
5 #ifdef COCCINELLE
6 @@
7 @@
8 - -= b = --a
9 #endif /* COCCINELLE */
Spatch (Editor #1)
Spatch
Compiler options..
1 <Compilation failed>
2
3 # For more information see the output window
4 # To open the output window, click or drag th
Output (0/6)
Spatch
- 102ms


COMPILER EXPLORER
Add...
More
Templates
Share
Other
C with
C with Coccinelle
Spatch (Editor #1)
Spatch
options...
Output of Spatch (Compiler #1)
Wrap lines
Select all
init_defs_builtins: /usr/local/lib/coccinelle/standard.h
minus: parse error:
File "patch.cocci", line 3, column 6, charpos = 12
around = '-=',
whole content = - -= b = --a
Compiler returned: 255
<Compilation failed>
For more information
To open the output wi
Output (0/0)
Spatch
57ms


COMPILER EXPLORER
DEV
Add...
More
Templates
Share
Other
C++ with Coccinelle source #1
C++ with Coccinelle
Spatch (Editor #1)
Spatch
Compiler options..
1 int main(){
2 a -= b = --a +1;
3 }
4
5 #ifdef COCCINELLE
6 @@
7 @@
8 - a -= b = --a +1;
9 + a -= +1;
10 #endif /* COCCINELLE */
1 int main(){
2 a -= +1;
3 }
Output (13/0)
Spatch
- 103ms (29B)


COMPILER EXPLORER DEV
Add...
More
Templates
Share
Other
C++ with Coccinelle source #1
C++ with Coccinelle
Spatch (Editor #1)
Spatch
Compiler options...
struct S { int operator [] (int
int main()
{
int a[1][1][1];
int i=0,j=0,k=0;
return a[i][j][k];
}
#ifdef COCCINELLE
@@ @@
int main() {
- int a[1][1][1];
+ S a;
...
}
@@
symbol a,i,j,k;
@@
- a[i][j][k]
+ a[i, j, k]
#endif /* COCCINELLE */
struct S { int operator [] (int x,int y,int z) {ret
int main()
{
S a;
int i=0,j=0,k=0;
return a[i, j, k];
}
Output (18/0)
Spatch
- 83ms (147B)


Add
More
Templates
Share
Other
Load and save editor text
Examples
Browser-local storage
Browser-local history
File system
Load/save to your system
Load from a local file
Save to file
Close
@@
symbol a,i,j,k;
@@
- a[i][j][k]
+ a[i, j, k]
Output (18/0)
Spatch
- 83ms (147B)
localhost:10240/#load-examples

How to run Coccinelle Explorer

1. Install Coccinelle locally (https://coccinelle.gitlabpages.inria.fr/website/)
2. Check https://github.com/compiler-explorer/compiler-explorer/pull/7001 and get the source code
3. As of now, Compiler Explorer requires specific version of Node.js; e.g. in `path-to-your/node-v20.18.2-linux-x64`
4. `EXTRA_ARGS=--language coccinelle_for_cpp,coccinelle_for_c NODE_DIR=path-to-your/node-v20.18.2-linux-x64 make dev`
5. Point browser at http://localhost:10240/

Perspectives

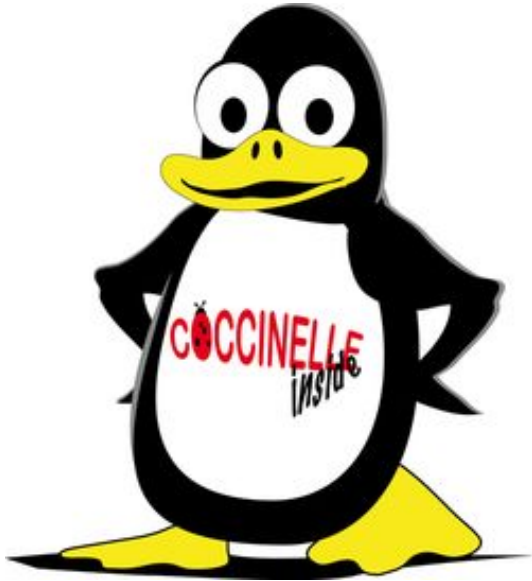
- proof of concept
- under review
  https://github.com/compiler-explorer/compiler-explorer/pull/7001
- not yet there:
  - separate windows for C/C++ and SmPL
  - specific material
  - which default options?

Perspectives

- proof of concept
- under review
  https://github.com/compiler-explorer/compiler-explorer/pull/7001
- not yet there:
  - separate windows for C/C++ and SmPL
  - specific material
  - which default options?
- feedback welcome!

Perspectives

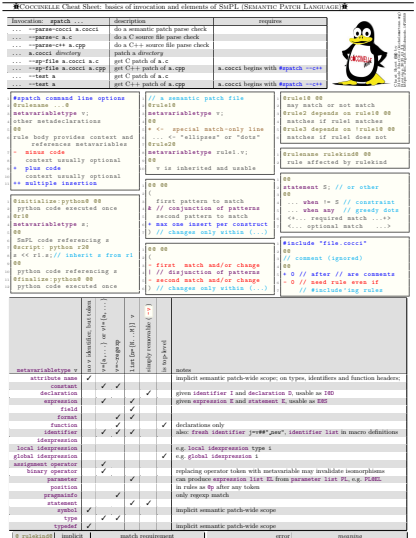
- proof of concept
- under review
  https://github.com/compiler-explorer/compiler-explorer/pull/7001
- not yet there:
  - separate windows for C/C++ and SmPL
  - specific material
  - which default options?
- feedback welcome (also about learning Coccinelle)!


COCCINELLE
inside

New: full-day Coccinelle training

- held at LRZ in Garching, Germany on 22.01.2025
- elements of grammar
- examples appendix, rather HPC-related
- https://doi.org/10.5281/zenodo.14728519 ⇓ lecture notes

New: cheat sheet



- selected essentials
- 1 page!
- https://doi.org/10.5281/zenodo.14728558 ⇓

Stay tuned

- We'll offer a Coccinelle training this year (not scheduled yet)
- LRZ trainings 2025 calendar: ⇓

https://tiny.badw.de/zc5D9e

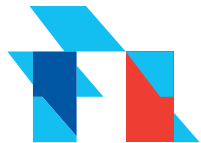
Work made possible by

SiVeGCS Project

GCS

Gauss Centre for Supercomputing

BayFrance23 travel grant



Thanks to Julia LAWALL for the collaboration!