Handling #ifdef Expressions in CPPSTATS

Claus Hunsen

University of Passau

September 2014

UNIVERSITÄT PASSAU

Software
Product-Line
Group

Outline

1 Introduction

2 Example

3 Source-Code Preparation Before Generating SRCML files

4 Processing of Expressions During File Analysis *

5 Global `#ifdef` Expression Pool *

6 Summary

Outline of the CPPSTATS Mechanisms

- Source-Code Preparation Before Generating SRCML files
  - Multi-Line #ifdef Expressions
  - Rewriting of #ifdef and #ifndef
  - Removal of Include Guards
- Processing of Expressions During File Analysis *
- Global #ifdef Expression Pool
  - Listing of #ifdef Expressions per File *
  - Construction of Global Expression Pool

* – affects scattering and tangling measurement over #ifdefs

Studies using CPPSTATS

This algorithm have been used in the following studies:

- Jörg Liebig, Sven Apel, Christian Lengauer, Christian Kästner, and Michael Schulze. *An Analysis of the Variability in Forty Preprocessor-Based Software Product Lines*. In ICSE, pages 105–114. ACM, 2010.
- Jörg Liebig, Christian Kästner, and Sven Apel. *Analyzing the Discipline of Preprocessor Annotations in 30 Million Lines of C Code*. In AOSD, pages 191–202. ACM, 2011.
- Claus Hunsen, Bo Zhang, Janet Siegmund, Christian Kästner, Olaf Leßenich, Martin Becker, and Sven Apel. *Preprocessor-Based Variability in Open-Source and Industrial Software Systems: An Empirical Study*. Empirical Software Engineering, 2014. Submitted.

Outline

1 Introduction

2 Example

3 Source-Code Preparation Before Generating SRCML files

4 Processing of Expressions During File Analysis *

5 Global #ifdef Expression Pool *

6 Summary

Example of Three Files

```
#ifdef A
  #ifdef B
  #endif
#endif
#ifdef A
#endif
```

(a) Nested #ifdef in file X.c.

```
#if defined(A) \
     && defined(B)
#else
#endif
```

(b) #else branch in file Y.c.

```
#ifndef Z_H
#define Z_H
#ifdef C
#elif defined(D)
#endif
#endif // Z_H
```

(c) #elif branch and include guard in file Z.h.

Figure: Short examples of the patterns that occur while using CPP and that are treated by CPPSTATS. Each example and their rewriting rules are explained in this very document.

Outline

1 Introduction

2 Example

3 Source-Code Preparation Before Generating SRCML files

4 Processing of Expressions During File Analysis *

5 Global #ifdef Expression Pool *

6 Summary

Multi-Line #ifdef Expressions

```
5 #if defined(A) \
6  && defined(B)
7 #else
8 #endif
```

(a) Repetition of Fig. 1b, containing a multi-line expression on Lines 5 and 6.

```
5 #if defined(A) && defined(B)
6
7 #else
8 #endif
```

(b) The same code with single-line expressions.

Figure: An #ifdef expression in file Y.c, (a) before and (b) after rewriting multi-line expressions into single-line fashion.

Rewriting of `#ifdef` and `#ifndef`

`#ifdef E` ⟶ `#if defined(E)`

`#ifndef F` ⟶ `#if !defined(F)`

Removal of Include Guards

```
 8  #ifndef Z_H
 9  #define Z_H
10  #ifdef C
11  #elif defined(D)
12  #endif
13  #endif // Z_H
```

(a) Include guard in file Z.h (Lines 8, 9, and 13).

```
 8
 9
10  #if defined(C)
11  #elif (!(defined(C))) && (defined(D))
12  #endif
13
```

(b) The example after removal of the include guard.

Figure: The file Z.h (a) before and (b) after removal of the include guard.

Preparation: Before and After

```
1  #ifdef A
2    #ifdef B
3    #endif
4  #endif
5  #ifdef A
6  #endif
```

```
5  #if defined(A) \
6       && defined(B)
7  #else
8  #endif
```

```
8  #ifndef Z_H
9  #define Z_H
10 #ifdef C
11 #elif defined(D)
12 #endif
13 #endif // Z_H
```

(a) Before preparation.

```
1  #if defined(A)
2    #if defined(B)
3    #endif
4  #endif
5  #if defined(A)
6  #endif
```

```
5  #if defined(A) && defined(B)
6
7  #else
8  #endif
```

```
8
9
10 #if defined(C)
11 #elif defined(D)
12 #endif
13
```

(b) After preparation.

Outline

1 Introduction

2 Example

3 Source-Code Preparation Before Generating SRCML files

4 Processing of Expressions During File Analysis *

5 Global #ifdef Expression Pool *

6 Summary

Processing Steps

- Look at each `#ifdef` expression.
- Rewrite if nested, `#elif`, or `#else` *

nested conjoin expressions from inner and outer `#ifdefs`
`#elif` conjoin expression with negation of all previous expressions
`#else` use conjunction of all previous expressions as negations

Processing #ifdef Expressions *

```
#if defined(A)
  #if defined(B)
  #endif
#endif
#if defined(A)
#endif
```

```
#if defined(A) && defined(B)

#else
#endif
```

```


#if defined(C)
#elif defined(D)
#endif

```

(a) Before processing.

```
#if defined(A)
  #if defined(A) && defined(B)
  #endif
#endif
#if defined(A)
#endif
```

```
#if defined(A) && defined(B)

#elif !(defined(A) && defined(B))
#endif
```

```


#if defined(C)
#elif (!(defined(C))) && (defined(D))
#endif

```

(b) After processing.

Outline

1 Introduction

2 Example

3 Source-Code Preparation Before Generating SRCML files

4 Processing of Expressions During File Analysis *

5 Global #ifdef Expression Pool *

6 Summary

Global #ifdef Expression Pool

- Local #ifdef Expression Pools *
  - Take each expression **only once** per file
  - Comparison via string equality
- Construction of Global Expression Pool

Local #ifdef Expression Pools *

```
#if defined(A)
#if defined(A) && defined(B)
#endif
#endif
#if defined(A)
#endif
```

```
defined(A)
defined(A) && defined(B)
~~defined(A)~~
```

```
#if defined(A) && defined(B)

#elif !(defined(A) && defined(B))
#endif
```

```
defined(A) && defined(B)
!(defined(A) && defined(B))
```

```


#if defined(C)
#elif (!(defined(C))) && (defined(D))
#endif

```

```
defined(C)
(!(defined(C))) && (defined(D))
```

Construction of Global Expression Pool

```
 defined(A)
defined(A) && defined(B)
```

```
 defined(A) && defined(B)
!(defined(A) && defined(B))
```

```
 defined(C)
(!(defined(C))) && (defined(D))
```

(a) Pools per file.

```
 defined(A)
defined(A) && defined(B)
defined(A) && defined(B)
!(defined(A) && defined(B))
defined(C)
(!(defined(C))) && (defined(D))
```

(b) Global expression pool.

Note

Scattering and tangling analyses are performed based on the global pool!

Outline

1 Introduction

2 Example

3 Source-Code Preparation Before Generating SRCML files

4 Processing of Expressions During File Analysis *

5 Global #ifdef Expression Pool *

6 Summary

Summary

- Processing of #ifdef expressions and construction of local expression pools (one per file) affect scattering and tangling analyses that work per #ifdef.
- Nesting analyses is not affected.